
# Identificación de sistemas estructurales histeréticos usando algoritmos de optimización multi-objetivo

**Gilberto Alejandro Ortiz García**

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica,
Electrónica y Computación
Manizales, Colombia
2013

# Identification of hysteretic structural systems using multi-objective optimization algorithms

**Gilberto Alejandro Ortiz García**

Dissertation submitted in partial fulfillment of the requirements for the degree of:
**Master of Engineering - Industrial Automation**

Adviser:
Dr. Techn. Diego Andrés Alvarez Marín
Co-Adviser:
Ph.D. Daniel Alveiro Bedoya–Ruiz

Research line:
Nonlinear structural system identification
Research group:
Earthquake Engineering and Seismology Group

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic
Engineering and Computing
Manizales, Colombia
2013

To my Family who has been always there to encourage me. Also I am grateful to my advisers Diego and Daniel who taught me another approach to civil engineering, and that all takes time, but every commitment can be accomplished with effort and a little stubbornness.

# Acknowledgments

I would like to show my gratitude to my advisers Diego and Daniel because they gave me support and continuous assistance, not only in the academic field, but also in my private life. I also want to thank my family, especially my mother for the confidence that she has relied on me, and her words that give me strength to keep on going in order to pursue my dreams.
Of course I also want to especially thank to:

- the kind people who have been around me. I would like to thank Felipe for his friendship and encouragement during this process.
- COLCIENCIAS for the support under the program "Young Researchers and Innovators - Virginia Gutiérrez de Pineda" (Code No. M00480990103846).
- Universidad Nacional de Colombia at Manizales for being my "Alma Mater" and the place where I learnt many things, not only from an academic viewpoint, but also from a personal and social perspective.
- the Earthquake Engineering and Seismology Group from Universidad Nacional de Colombia at Manizales
- all the people and institutions that contribute directly or indirectly with this research, my most sincere thanks.

Manizales, Colombia Gilberto A. Ortiz

April, 2013.

# Resumen

La mayoría de la literatura referente a la estimación de parámetros de los modelos de histéresis de tipo Bouc-Wen por medio de algoritmos evolutivos no solo usa una función objectivo única (el error cuadrático medio entre los desplazamientos conocidos y los estimados), sino que también considera el modelo original de histéresis de Bouc-Wen (sin degradación y sin pinching) en el proceso de identificación. En esta Tesis se presenta una metodología novedosa para la estimación de los parámetros del modelo de histéresis de Bouc-Wen-Baber-Noori. La metodología está basada en un algoritmo evolutivo para optimización multi-objetivo, llamado NSGA-II [34]; por lo tanto, un conjunto de funciones objetivo es empleado en vez de una función objetivo única. Esta metodología no solo minimiza la diferencia entre los desplazamientos medidos en el laboratorio y los estimados, sino que también minimiza la diferencia entre la energía disipada experimental y la estimada con el modelo. La metodología propuesta identifica el sistema estructural y permite la observación de la multi-modalidad del modelo de histéresis de BWBN. El desempeño del algoritmo es evaluado usando datos simulados y reales.

**Palabras claves: Modelo de Bouc-Wen-Baber-Noori; Histéresis; Optimización multi-objetivo; Identificación de sistemas; NSGA-II.**

# Abstract

Most of the published literature concerned with the parameter estimation of the Bouc-Wen model of hysteresis via evolutionary algorithms not only uses a single objective function (the mean square error between the known displacements and the estimated ones) but also considers the original Bouc-Wen model of hysteresis (without degradation and pinching) in the identification process. In this thesis, a novel method for the identification of the parameters of the Bouc-Wen-Baber-Noori (BWBN) model of hysteresis is presented. The methodology is based on a multi-objective evolutionary optimization algorithm called NSGA-II [34]; therefore, a set of objective functions is employed instead of a single objective function. The methodology minimizes not only the difference between the measured displacements at the laboratory and the estimated ones, but also minimizes the difference between the experimental dissipated energy and the estimated one. The proposed methodology identifies the structural system and allows the observation of multi-modality of the BWBN model of hysteresis. The performance of the algorithm is evaluated using simulated and real data.

**Keywords: Bouc-Wen-Baber-Noori model; Hysteresis; Multi-objective optimization; System identification; NSGA-II.**

# Contents

# Conventions and Notation

## General notation

| Symbol | Meaning |
|---|---|
| $\boldsymbol{a}, \boldsymbol{b}, \ldots$ | Vector notation (Column vector) |
| $\boldsymbol{A}, \boldsymbol{B}, \ldots$ | Matrix notation |
| $\boldsymbol{\emptyset}$ | Null matrix/vector |
| $\mathscr{K}\{\boldsymbol{x}(t)\}$ | $\mathscr{K}$–transform of $\boldsymbol{x}(t)$ |
| $\mathscr{F}\{\cdot\}$ | Fourier transform |
| $\mathscr{L}\{\cdot\}$ | Laplace transform |

## Dynamic of structures

| Symbol | Meaning |
|---|---|
| $\alpha$ | Rigidity ratio (Bouc–Wen model) |
| $\beta$ | Hysteresis shape parameter (Bouc–Wen model) |
| $\gamma$ | Hysteresis shape parameter (Bouc–Wen model) |
| $n$ | Hardening/Softening parameter (Bouc–Wen model) |
| $A(t)$ | Degradation function (Bouc–Wen–Baber–Noori model) |
| $A_0$ | Hysteresis amplitude (Bouc–Wen–Baber–Noori model) |
| $\delta_A$ | Hysteresis amplitude parameter (Bouc–Wen–Baber–Noori model) |
| $\delta_\eta$ | Stiffness degradation parameter (Bouc–Wen–Baber–Noori model) |
| $\delta_\nu$ | Strength degradation parameter (Bouc–Wen–Baber–Noori model) |
| $\varepsilon(t)$ | Absorbed hysteretic energy |
| $\eta(t)$ | Stiffness degradation function (Bouc–Wen–Baber–Noori model) |

| Symbol | Meaning |
|---|---|
| $\eta_0$ | Stiffness degradation (Bouc–Wen–Baber–Noori model) |
| $h(t)$ | Pinching function (Bouc–Wen–Baber–Noori model) |
| $\nu(t)$ | Strength degradation function (Bouc–Wen–Baber–Noori model) |
| $\nu_0$ | Strength degradation (Bouc–Wen–Baber–Noori model) |
| $\lambda$ | Pinching parameter (Bouc–Wen–Baber–Noori model) |
| $p$ | Initial pinching (Bouc–Wen–Baber–Noori model) |
| $\psi_0$ | Pinching parameter (Bouc–Wen–Baber–Noori model) |
| $\delta_\psi$ | Change of pinching (Bouc–Wen–Baber–Noori model) |
| $q$ | Pinching parameter (Bouc–Wen–Baber–Noori model) |
| $\varsigma_0$ | Total slip (Bouc–Wen–Baber–Noori model) |
| $\varsigma_1(t)$ | Pinching parameter (Bouc–Wen–Baber–Noori model) |
| $\varsigma_2(t)$ | Pinching parameter (Bouc–Wen–Baber–Noori model) |
| $z_u$ | Ultimate value of $z(t)$ (Bouc–Wen–Baber–Noori model) |
| $c$ | Linear viscous damping coefficient |
| $c_c$ | Critical damping |
| $\xi$ | Viscous damping ratio |
| $p(t)$ | External excitation |
| $F_D(t)$ | Damping force |
| $F_I(t)$ | Inertial force |
| $F_R(t)$ | Restoring force (linear structural systems) |
| $F_H(\dot{x}(t), x(t); t)$ | Restoring force (nonlinear structural systems) |
| $F_T(x(t), z(t); t)$ | Non–damping restoring force (Bouc–Wen model) |
| $k$ | Elastic stiffness |
| $k_f$ | Post-yield stiffness |
| $k_i$ | Pre-yield stiffness |
| $m$ | Mass |
| $x(t)$ | Displacement |
| $\dot{x}(t)$ | Velocity |
| $\ddot{x}(t)$ | Acceleration |
| $\ddot{x}_g(t), u(t)$ | Ground acceleration |

| Symbol | Meaning |
|---|---|
| $z(t)$ | Hysteretic displacement |
| $\omega_0$ | Pseudo-natural frequency |

## Evolutionary Algorithms

| Symbol | Meaning |
|---|---|
| $\beta_{qi}$ | Spread factor (real–parameter GA) |
| $\bar{\delta}$ | Mutation parameter (real–parameter GA) |
| $\eta_c$ | Crossover distribution index (real–parameter GA) |
| $\eta_m$ | Mutation distribution index (real–parameter GA) |
| $\boldsymbol{f}(\boldsymbol{x}(t))$ | Fitness function of individual $\boldsymbol{x}(t)$ |
| $\boldsymbol{F}(t)$ | Fitness of the whole population $\boldsymbol{P}(t)$ |
| $\boldsymbol{x}(t)$ | Individual at time $t$ |
| $\boldsymbol{x}'(t)$ | Individual at time $t$ after recombination |
| $\boldsymbol{x}''(t)$ | Individual at time $t$ after mutation |
| $l$ | Length of the binary string (genetic algorithms) |
| $\boldsymbol{x}^l$ | Lower boundary population $\boldsymbol{P}(t)$ |
| $\boldsymbol{x}^u$ | Upper boundary population $\boldsymbol{P}(t)$ |
| $\mu$ | Parent population size |
| $\theta$ | Offspring population size |
| $\boldsymbol{P}(t)$ | Population at time $t$ |
| $\boldsymbol{P}'(t)$ | Population at time $t$ after recombination |
| $\boldsymbol{P}''(t)$ | Population at time $t$ after mutation |
| $p_c$ | Crossover probability |
| $p_m$ | Mutation probability |
| $\chi$ | Random number |
| $\alpha_{ij}$ | Rotation angle (ES) |
| $\boldsymbol{A}$ | Matrix of rotation angles (ES) |
| $\boldsymbol{\sigma}$ | Vector of standard deviations (ES) |
| $\boldsymbol{\Sigma}$ | Covariance matrix (ES) |

| Symbol | Meaning |
|---|---|
| $n_\sigma$ | Number of standard deviations (ES) |
| $n_\alpha$ | Number of rotation angles (ES) |
| $\mathcal{D}$ | Decision space or parameter space |
| $\mathcal{Z}$ | Objective space |
| $\Theta_m$ | Mutation operator |
| $\Theta_r$ | Recombination operator (Crossover) |
| $\Theta_s$ | Selection operator |
| $\beta$, $\tau$, $\tau'$ | Learning rates (ES) |
| $\boldsymbol{R}(\alpha_{ij})$ | Rotation matrix (ES) |
| $\boldsymbol{S}\,(\sigma_{ii})$ | Diagonal matrix of standard deviations (ES) |
| $\boldsymbol{T}\,(\alpha_{ij})$ | Orthogonal rotation matrix (ES) |

# List of Acronyms

| Acronym | Meaning |
|---|---|
| BWBN | Bouc–Wen–Baber–Noori |
| EA | Evolutionary Algorithm |
| EKF | Extended Kalman Filter |
| ES | Evolution Strategies |
| DE | Differential Evolution |
| GA | Genetic Algorithm |
| LTI | Linear Time Invariant |
| MDOF | Multiple Degrees Of Freedom |
| MOBEA | Multi–Objective Evolutionary Algorithm |
| MOO | Multi–objective Optimization |
| PF | Particle Filter |
| PSO | Particle Swarm Optimization |
| SDOF | Single Degree Of Freedom |
| UKF | Unscented Kalman Filter |

# 1. Introduction

## 1.1. Motivation

The identification of nonlinear hysteretic systems has been an important topic in structural dynamics over the last decades, because many practical applications encountered in civil and mechanical engineering are related to the modeling and identification of this type of systems. In the field of structural analysis and design, it is mandatory to use a model that predicts the behaviour of structures under the action of dynamic loads, as these are the ones that usually determine the designs; the structures are projected in such a way that they have an appropriate performance against dynamic phenomena such as earthquakes or strong winds. Structures subjected to time-varying loads are designed so that they are able to dissipate the energy generated by the excitation by means of the inelastic behaviour of the materials that comprise the system, however, the repeated action of these loads will cause the deterioration of the material, and in consequence each load cycle will give different displacement levels that will be conditioned by the actual state of the material.

In this way, the engineer must use models that allow him to identify the parameters that characterize the degradation of the system under study; such models must be able to take into account not only the instantaneous displacements that occur in the structure, but also the past history that has been experienced by the structure.

When considering the *memory* of the material, the model complexity increases, and sometimes a nonlinear model is required. This kind of models require different methods of analysis and identification than the ones used in the case of linear systems.

The Bouc-Wen-Baber-Noori (BWBN) model of hysteresis is very popular in the structural dynamics field, given its ability to model common physical phenomena that occur when dynamic loads are exerted on the structures. This characteristic allows the model to describe a wide variety of hysteresis cycles. Several techniques have been developed in order to identify the complete set of parameters of the BWBN model of hysteresis, so that an approximation can be given as accurate as possible and that the resulting model can be used to identify basic characteristics of structural systems subjected to random vibrations.

Several methodologies used for the identification of the BWBN model of hysteresis are based on *nonlinear filtering* methods (e.g. Kalman filters [101, 96, 25], particle filters [66, 25]), which from probability distributions let us estimate the values of parameters within the model. Another group of techniques use *optimization* procedures (e.g., Gauss-Newton methods [101], Evolutionary Algorithms [24, 43], Particle Swarm Optimization [100, 23]), which

are methods based on the minimization of a loss function.

## 1.2. Problem statement

We want to develop a methodology for the identification of structural systems that exhibit nonlinear hysteretic behaviour in order to model as precisely as possible the dynamic behaviour of the system so that a reliable design of the structure can be done thus reducing the uncertainty generated by the model.

## 1.3. Objective

The main idea of this document is to provide a novel methodology for the identification of hysteretic structural systems using multi-objective optimization techniques based on genetic algorithms. Up to the author's knowledge, all of the proposed methods for the identification of the parameters of the BWBN model just try to minimize the mean square error between the estimated displacements and the ones measured in the real structure. In this thesis, it will be proposed a novel identification method that simultaneously minimizes not only the dissimilarities between the estimated displacements and the real ones, but also between the dissipated energy by the structure and the one estimated using the mathematical model, so a better approximation of the hysteretic behaviour is expected.
Given this main idea, the following specific objectives are proposed:

- Implement a multi-objective optimization algorithm in MATLAB® and C in order to provide a toolbox for the identification of the parameters of the BWBN model of hysteresis.
- Perform simulations in order to evaluate the performance of the proposed methodology. The data used for the simulations are provided by Prof. Bedoya-Ruíz, and they correspond to an experimental campaign carried out on ferrocement walls tested on a reaction wall.
- Publication of articles in high impact journals and conference proceedings.

## 1.4. Outline of the thesis

The idea is to make this thesis as self-contained as possible, so Chapter 2 is devoted to the theory of dynamic of structures and hysteresis modeling, so a better understanding of the terms, model and methods used through the document is expected. Chapter 3 makes a brief introduction to the theory of optimization and Chapter 4 aims to provide the theory related with evolutionary algorithms, which are the methods used in this thesis in order to

identify the parameters of the BWBN model of hysteresis. Chapter 5 introduces the idea of multi-objective evolutionary algorithms, and shows how the theory of single objective optimization techniques is extended in order to handle more than one objective function. Chapter 6 gives a brief description of the methods found in the literature that are used to identify the parameters of the BW-type models of hysteresis. Chapter 7 presents a novel methodology for the identification of hysteretic structural systems that exhibit not only degradation but also pinching. The document ends with a summary of the main results found in this research and offers suggestions for future work. Additionally, there are two appendices where artificial landscapes used to test the algorithms developed in this thesis are presented, and the final research products are listed.

# Part I.

# Foundations

# 2. An overview of structural dynamics

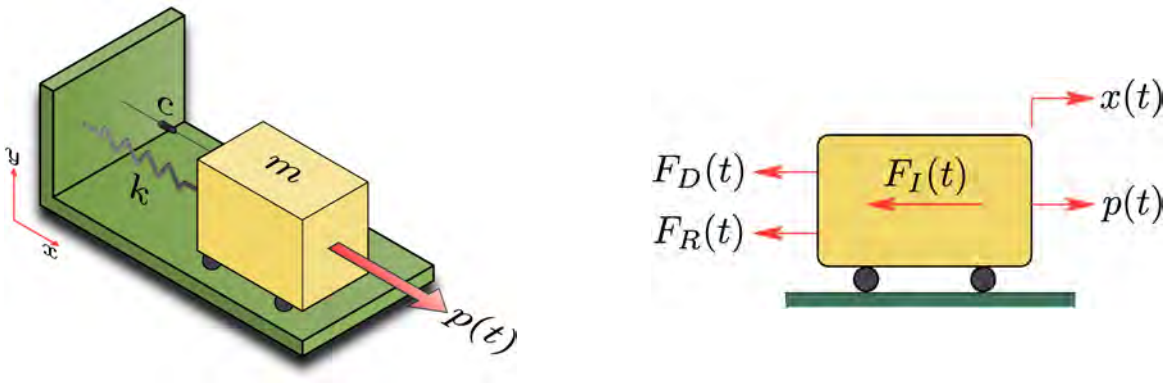
Dynamic of structures is the branch of structural analysis that studies the effects of external excitations that produce vibrations in structures.
This introductory Chapter to dynamic of structures starts with the analysis of *linear* and *nonlinear* structural systems; all the theory developed herein is for *single degree of freedom* (SDOF) systems, which are systems whose components can move in only one direction, while the other directions are restricted. Analysis of SDOF systems provide the basis for the understanding and treatment of the vast majority of structural dynamic problems, the generalization to *multiple degrees of freedom* (MDOF) systems is possible; for more information about such generalization, the reader is referred for example to [28, 30, 51]. The Chapter also gives an introduction to the phenomenon of hysteresis, which is a nonlinear behaviour very common in structural elements subject to time-varying loads; then, a brief description of existing hysteresis models is presented; special attention is given to the Bouc–Wen model of hysteresis and its generalizations, inasmuch as such models are widely used in the analysis of hysteretic behaviour in mechanical engineering and active control of civil infrastructure, and because it is the model used throughout this work.

## 2.1. Dynamic response of linear structural systems

In dynamical analysis, the response of a linear structural system is obtained from the application of an external excitation, whose temporal variation is known at each time instant. The *response* of the system is typically a record of displacements that occurs in the structure. The mathematical representation of a linear SDOF system subjected to an external excitation is given by the *equation of motion*, which is derived from Newton's second law ($F = ma$). The idealization of this system comprises a mass ($m$), a *viscous damping* mechanism with damping constant $c$, and a spring with stiffness $k$ (see Fig. **2-1**(a)).
The restoring forces that act on the body are shown in Fig. **2-1**(b). Mathematically, the equation of movement for the system in Fig. **2-1** is given by:

$$F_I(t) + F_D(t) + F_R(t) = p(t), \tag{2-1}$$

where $F_I(t) = m\ddot{x}(t)$ is the *inertial force*, $F_D(t) = c\dot{x}(t)$ is the *damping force*, $F_R(t) = kx(t)$ is the *restoring force*, $p(t)$ is the *external excitation* acting on the mass and $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ are the displacement, velocity and acceleration of the mass respectively. The substitution



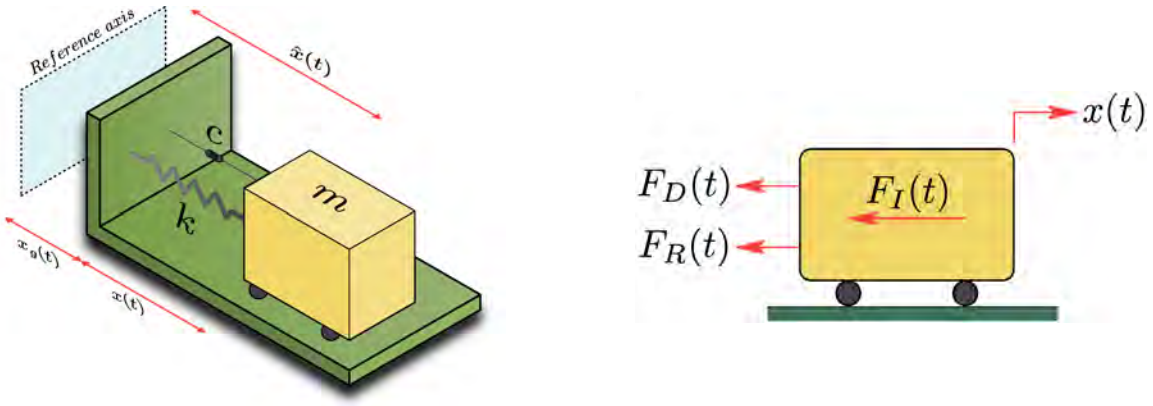

(a) Idealization (b) Free body diagram

**Figure 2-1**.: Single degree of freedom system

of these values of the restoring forces in Eq. (2-1) gives the equation of motion for a linear SDOF system:

$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = p(t).$

For the case of seismic analysis, there is not an external excitation acting directly on the structure (see Fig. **2-2**)




(a) Response of the system (b) Free body diagram

**Figure 2-2**.: Influence of support excitation

In earthquake engineering, a SDOF structure under seismic load is represented by such a model. Following a similar procedure as above, the equation of motion of the system in Fig. **2-2** is given by:

$$F_I(t) + F_D(t) + F_R(t) = 0, \tag{2-2}$$

where the damping force and the restoring force are the same as in Eq. (2-1), however, the inertial force is different in this case, and is given by:

$F_I(t) = m\ddot{\hat{x}}(t),$

where $\hat{x}(t)$ is the total displacement experienced by the structure. Such displacement can be decomposed into the sum of the ground motion $x_g(t)$ and the displacement of the structure due to the deflection of its components $x(t)$. Therefore, the net acceleration experienced by the structure is $\ddot{x}_g(t) + \ddot{x}(t)$. Substituting all the terms in Eq. (2-2) give the linear model of a SDOF system subjected to seismic load (see e.g. [10]):

$$\begin{aligned} m\left[\ddot{x}_g(t) + \ddot{x}(t)\right] + c\dot{x}(t) + kx(t) &= 0, \\ m\ddot{x}(t) + c\dot{x}(t) + kx(t) &= -m\ddot{x}_g(t). \end{aligned} \tag{2-3}$$

Eq. (2-3) is generally expressed as:

$$\begin{aligned} \ddot{x}(t) + \frac{c}{m}\dot{x}(t) + \frac{k}{m}x(t) &= -\ddot{x}_g(t), \\ \ddot{x}(t) + 2\xi\omega_0\dot{x}(t) + \omega_0^2 x(t) &= -\ddot{x}_g(t), \end{aligned} \tag{2-4}$$

with initial conditions $\dot{x}(0) = v_0$ and $\ddot{x}(0) = a_0$. Here, $v_0$ and $a_0$ represent the initial velocity and acceleration of the system, $\omega_0$ is the *pseudo-natural frequency* of the structure, which is an intrinsic property of the system given by:

$$\omega_0 = \sqrt{\frac{k}{m}},$$

and $\xi$ is the *viscous damping ratio* given by:

$$\xi = \frac{c}{c_c} = \frac{c}{2m\omega_0},$$

where $c_c$ is defined as the *critical damping* because it defines the boundary between damped and undamped vibrations.

With the model given by Eq. (2-4), it is possible to compute the responses of the system (displacements and accelerations of the structure), which are the most important quantities used by engineers in the design of buildings because both results are related to other analysis such as *structural reliability, structural safety* and *structural serviceability.*

## 2.2. Dynamic response of nonlinear structural systems

For the case of nonlinear structural systems, the theory described in Section 2.1 is not applicable because the stiffness and the damping of the system are no longer proportional to the displacement and velocity of the mass respectively. The new model used for the analysis of nonlinear systems is more complex and requires numerical procedures in order to compute the displacements.

In structural dynamics, nonlinear systems subjected to a seismic load have a response given by the following differential equation:

$$m\ddot{x}(t) + F_H(\dot{x}(t), x(t); t) = -m\ddot{x}_g(t), \tag{2-5}$$

where $F_H(\dot{x}(t), x(t); t)$ is the *restoring force* of the nonlinear structural system, which depends on the velocity and displacement of the structure.

One of the nonlinear behaviour experienced by systems under severe excitations is known as *hysteresis*; it appears when the structural response becomes inelastic. Such behaviour is very common because no material used in mechanical and structural engineering is perfectly elastic, hence the restoring forces generated by deformations are not conservative [46]; this is reason why hysteresis is seen as a natural mechanism developed by the materials to dissipate energy generated by the external excitation [53]; such characteristic is exploited in the design of structural members or connections in order to increase the margin of safety against external dynamic loads.

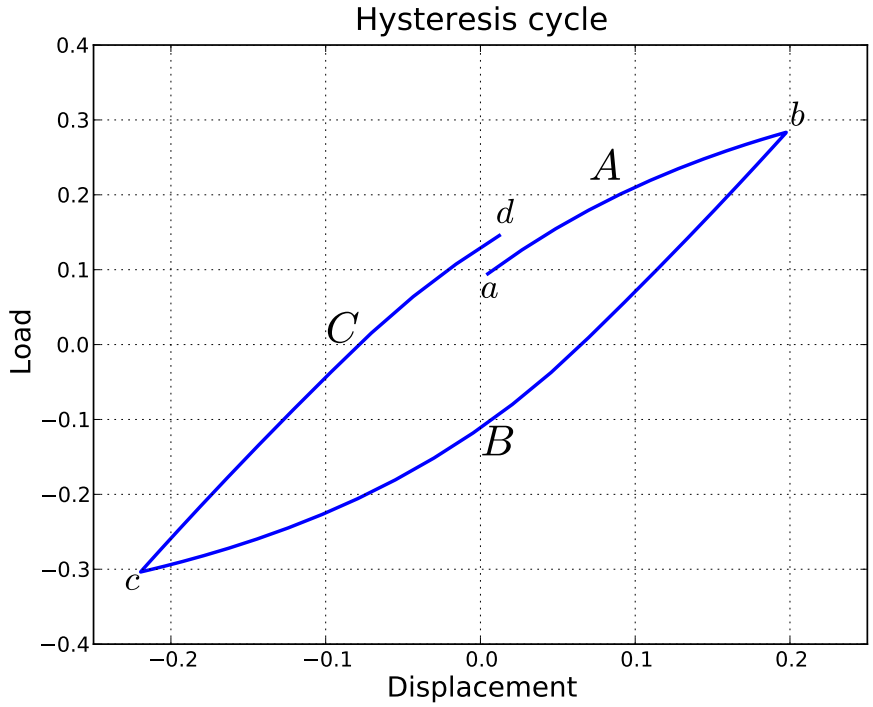



**Figure 2-3**.: Hysteresis loop

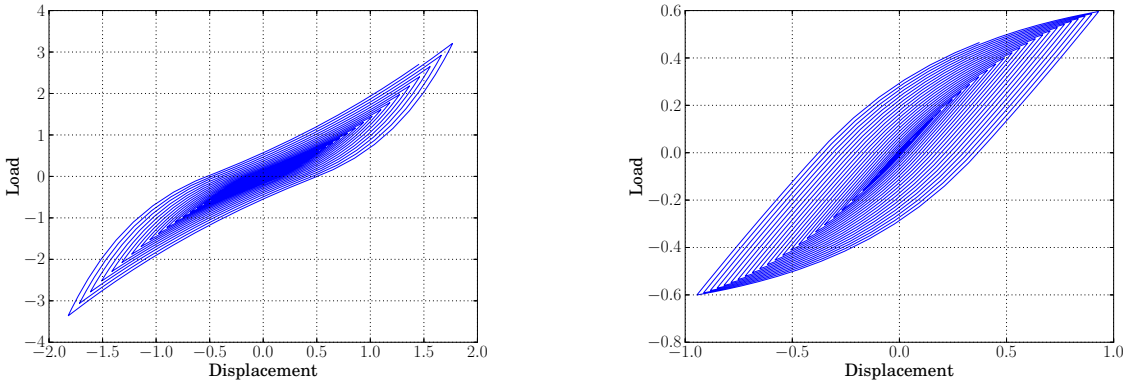
Fig. **2-3** is the typical representation of hysteretic behaviour in a force vs. displacement curve. It can be seen that the first branch (branch $A$) starting at point $a$ represents a *loading* stage; the body is under an increasing load, therefore the body suffers a deformation. When the system reaches point $b$, a reversal load is applied, and the mass tries to recover its original state, but since the applied force exceeded the *elastic range* of the material, the body suffered a permanent deformation, and that is why branch $B$ does not return by the same path that branch $A$. This process continues until the external excitation ceases or the material fails. This basic concept is the main idea underlying actual performance–based approaches in structural design.

Nowadays, structural engineering community has noticed the importance of evaluating the structural behaviour beyond the onset of damage, and that is the reason why structural design is moving towards *fragility analysis*. The primary objective is the development of accurate and reliable techniques capable of predict quantitatively the level of damage suffered by any structure subjected to dynamical loads. Then, computer programs with the implementation of such models are needed to perform nonlinear structural analysis faster [86].

### 2.2.1. Physical behaviour of structural systems when subjected to dynamic loads

Following, a description of some characteristics of structural systems when subjected to dynamical loads are defined. Basically, these phenomena are due to a progressive deterioration of the material, and also due to the interaction between constitutive elements of the system.

- **Hardening:** Increase of the stiffness of the system. In a hysteresis curve, hardening is related with an increase in the slope of such curve with the increase in displacement (see Fig. **2-4**(a)).
- **Softening:** Reduction of the stiffness of the system. In a hysteresis curve, softening is related with a decrease in the slope of such curve with the increase in displacement (see Fig. **2-4**(b)).

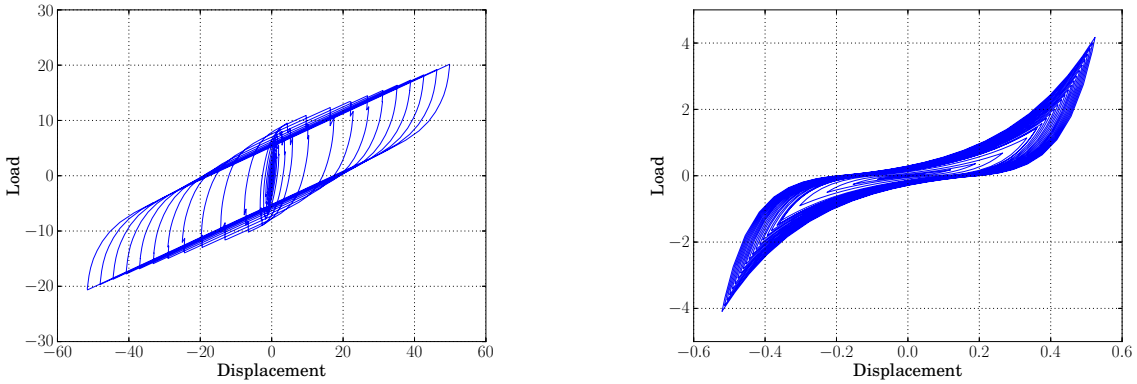



(a) Hardening (b) Softening

**Figure 2-4**.: Effects of hardening and softening in hysteresis cycles

- **Strength degradation:** After each load cycle, the material diminishes its capacity to resist external loads (see Fig. **2-5**(a)).
- **Stiffness degradation:** The material experiences a progressive loss of stiffness after each load cycle (see Fig. **2-5**(a)).

- **Pinching:** Is a sudden loss of stiffness, followed by a rapid increase of it, often related with the interaction between structural elements. For instance, in concrete, pinching is induced by opening/closing of cracks; in wood systems, this phenomenon is inflicted by slipping at joints, etc (see Fig. **2-5**(b)).




(a) Strength and stiffness degradation (b) Pinching

**Figure 2-5**.: Effects of strength-stiffness degradation and pinching in hysteresis cycles

## 2.2.2. Hysteresis models

In the following, a brief description of several hysteresis models used in the analysis of nonlinear behaviour of structures subjected to time-varying loads is presented.

### Preisach model of hysteresis

This model was proposed by Preisach [79] and is widely used in the field of electrical engineering in order to simulate the hysteretic behaviour of ferromagnetic materials [61], but it has been extended so other phenomena can be modelled as well; for example, the model has been used to represent thermostats in a heat conduction control problem [21] and to model the hysteretic behaviour of mesoscopic materials [27], which are aggregates of grains that act as rigid vibrating bodies.
The Preisach model of hysteresis is based on the assumption that the hysteresis loops can be represented as a sum of an infinite number of weighted elementary interacting fragments, called *hysterons*, which are elementary hysteresis operators represented as rectangular loops. The resultant approximation of the hysteresis cycles is given by a staircase function as shown in Fig. **2-6**:
The approximation given by this model of hysteresis depends on the number of hysterons used, i.e., the model will fit the experimental hysteresis cycles with a good accuracy when a higher number of hysterons are used within the formulation. This is the main disadvantage of the Preisach model, because generally a large number of hysterons are required in order

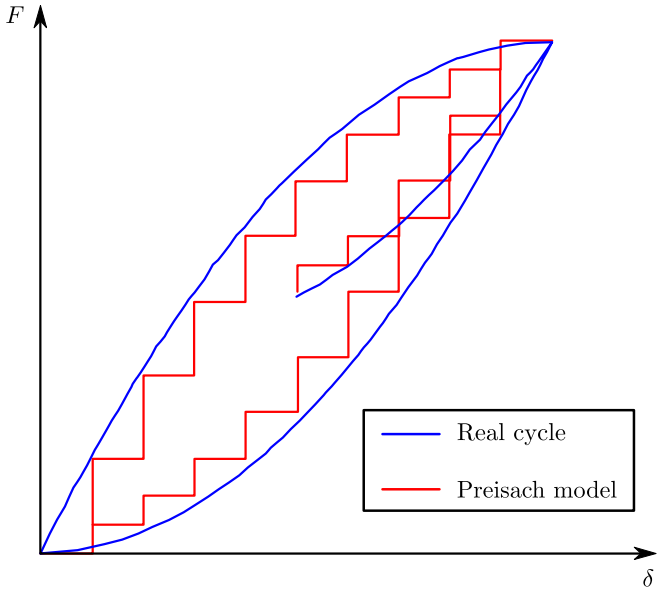



**Figure 2-6.**: Preisach model of hysteresis

to model complex behaviours such as degradation and pinching, but this implies a higher computational burden, and other models can achieve the same accuracy with less resources. Given the low popularity of this model in the field of mechanical engineering, we do not go further with this model. The reader interested in delving into this model is referred to [79] and [61], where the mathematical description of the model is presented, as well as other applications of the model.

### Ramberg–Osgood model

This algebraic model was proposed by Ramberg and Osgood [80]. This model has the ability to simulate gradual stiffness and strength degradation of the material. It has been used mainly in the modelling of hysteresis of members made of metals that suffer plastic deformations and hardening [11] and dampers made of low-yield steel [2].
The model takes each hysteresis cycle and computes an *skeleton curve* (see Fig. **2-7**) according to the following expression:

$$\frac{\varepsilon}{\bar{\varepsilon}_0} = \frac{\sigma}{\bar{\sigma}_0}\left(1 + \alpha\left|\frac{\sigma}{\bar{\sigma}_0}\right|^{\eta-1}\right),$$

where $\sigma$ is the stress, $\bar{\sigma}_0$ is the yield stress of the cycle being analyzed, $\varepsilon$ is the strain, $\bar{\varepsilon}_0$ is the yield strain of the cycle being analyzed, and $\alpha$ and $\eta$ are coefficients that define the shape of the hysteresis curve. The hysteresis loop given by the model has the same shape of the skeleton curve, but enlarged twice in both directions (horizontally and vertically).

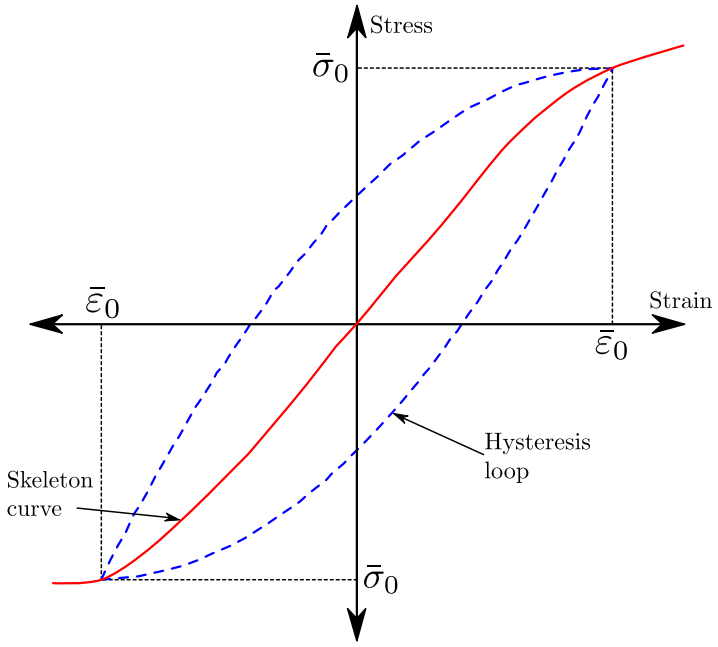



**Figure 2-7**.: Ramberg-Osgood model

If the coefficient $\eta$ approaches to infinity, the plastic deformation is zero when $\sigma < \bar{\sigma}_0$, is determinate when $\sigma = \bar{\sigma}_0$, and is infinite when $\sigma > \bar{\sigma}_0$ (impossible case). This limiting case describes a *perfectly plastic* solid with yield stress $\bar{\sigma}_0$.
The Ramberg-Osgood model is particularly well-suited to represent the virgin curve of work-hardening solids, especially ones without a sharply defined yield stress [68].

### Polygonal hysteretic models

These models are represented with piecewise linear functions that describe the behavioral stages of a material or structure such as elastic stage, stiffness and strength degradation, yielding, cracking, etc.
Polygonal hysteresis models are also referred to as multi-linear models. Several models of this type have been proposed in the literature [29, 40, 4] and have been used to model the hysteretic behaviour of concrete structures, plywood panels, wood structures, steel members, etc. [52]. The reader is referred to [76] and [86], where a comparative study of these models is presented.
The polygonal hysteresis models basically define a *backbone curve* (see Fig. **2-8**) which defines the increasing deformation experienced by the structure.
The shape of the backbone curve is given by the following set of parameters:

- $K_e$, the elastic stiffness.
- $F_y$ the yield strength.

- $K_s = \alpha_s K_e$, the strain-hardening stiffness.
- $\delta_y$, the yield deformation.
- $\delta_c$, the cap deformation.
- $F_c$, the peak strength of the hysteresis curve.
- $K_c = \alpha_c K_e$, the post-capping stiffness (usually has negative value).
- $F_r = \lambda F_y$ the residual strength, that is the fraction of yield strength that is preserved when a given deterioration stage is achieved.
- $\delta_r$, the residual deformation.

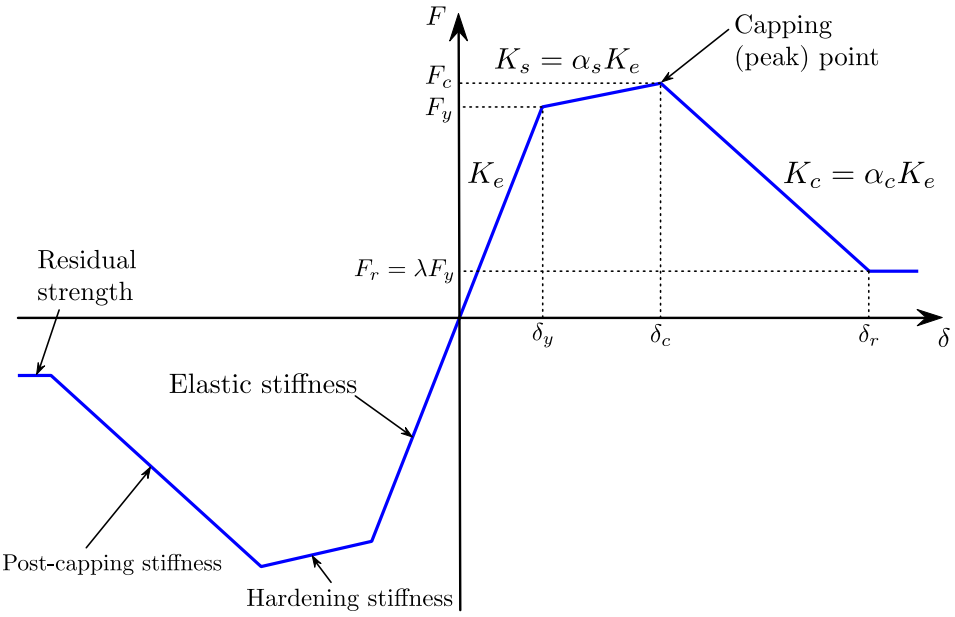



**Figure 2-8**.: Polygonal hysteresis model (adapted from [52])

The parameters $\alpha_s, \alpha_c$ and $\lambda$ are estimated, or they can be obtained through calibration of the model [52]. When there is no deterioration, the first three parameters determine completely the backbone curve, and the other parameters are neglected. In order to model complex behaviours (such as degradation or pinching), all the parameters defined above are used. Fig. **2-9** shows an example of a hysteresis cycles computed using a multi-linear polygonal model.

### Bouc-Wen type models of hysteresis

The Bouc-Wen (BW) model has been widely employed in structural engineering for the analytical description of the hysteretic behaviour given its capability to describe several patterns

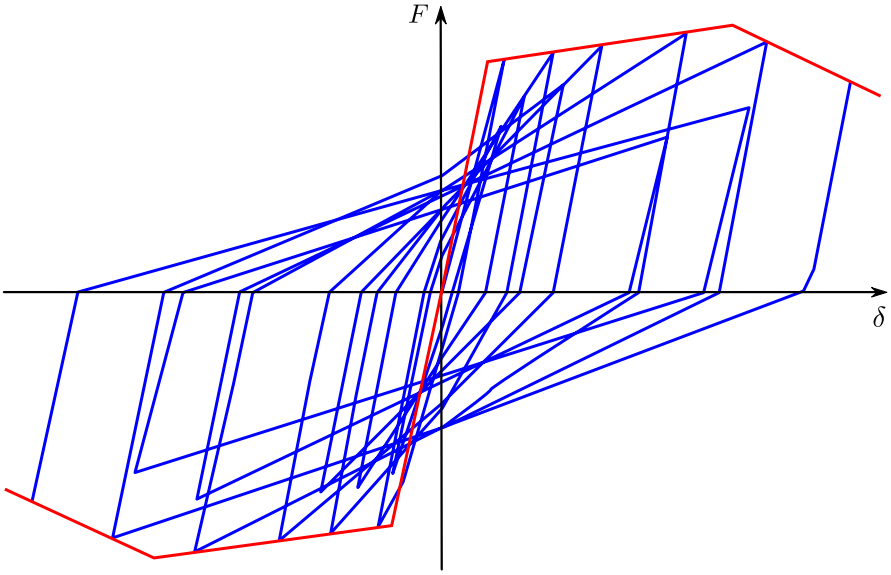



**Figure 2-9**.: Multi-linear polygonal model

that match the response of a wide class of hysteretic systems. The model was introduced by Bouc [19, 20] and generalized by Wen [93], with the advantage of computational simplicity, because only one auxiliary nonlinear differential equation is needed to describe the hysteretic behaviour. This model has been extended to describe various characteristics of hysteretic behaviour, like stiffness and strength degradation [7, 6, 5], pinching effect (see e.g. [38, 37]), biaxial hysteresis [77], asymmetry of the peak restoring force [91, 87], among others, leading to a broader class of models, henceforth named *BW-type models*.

Bouc-Wen type models have been used in a wide range of applications such as vibration of steel structures [73, 67], concrete structures [92, 62], wood joints [38, 1] or base isolation devices for buildings [54]. It has been also used to model magnetorheological dampers [9, 63], piezoelectric elements [44], and soil dynamics [78, 42], among others.

In the following we will introduce a generalized BW-type model that is usually known in the literature as the *Bouc-Wen-Baber-Noori (BWBN) hysteretic model* [7, 6], that is popular because it models not only degradation but also pinching, and which is the model addressed in this work. This hysteresis model considers the following nonlinear restoring force for Eq. (2-5):

$$F_H(\dot{x}(t), x(t); t) = c\dot{x}(t) + F_T(x(t), z(t); t), \tag{2-6}$$

which consists of a damping restoring force $F_D(t) = c\dot{x}(t)$ and a *non–damping restoring force* $F_T(x(t), z(t); t)$ given by:

$$F_T(x(t), z(t); t) = \alpha k_i x(t) + (1 - \alpha)\, k_i z(t),$$

where $\alpha k_i x(t)$ is the *linear restoring force* and $(1-\alpha)\, k_i z(t)$ is the *hysteretic restoring force*. The new variable $\alpha$ is the ratio of post-yield $k_f$ to pre-yield (elastic) $k_i$ stiffness:

$$\alpha = \frac{k_f}{k_i}; \qquad 0 < \alpha < 1, \tag{2-7}$$

and $z(t)$ is known as the *hysteretic displacement*.

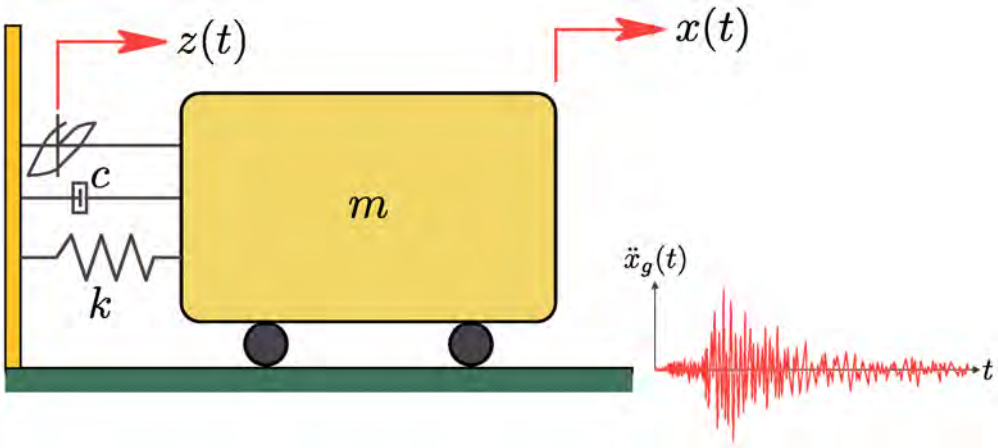



**Figure 2-10**.: Hysteretic SDOF system

Consider the hysteretic SDOF system subjected to seismic load, shown in Fig. **2-10**. The equation of motion of such system is given by Eq. (2-5) and Eq. (2-6):

$$m\ddot{x}(t) + c\dot{x}(t) + \alpha k x(t) + (1-\alpha)\, k z(t) = -m\ddot{x}_g(t),$$

dividing the previous equation by $m$, an analogous expression to Eq. (2-4) is derived:

$$\ddot{x}(t) + 2\xi\omega_0\dot{x}(t) + \alpha\omega_0^2 x(t) + (1-\alpha)\,\omega_0^2 z(t) = u(t). \tag{2-8}$$

Eq. (2-8) is the general representation of the equation of motion for a hysteretic SDOF system. It is worth noting that $u(t) = -\ddot{x}_g(t)$ can be any mass-normalized force function. The hysteretic displacement $z(t)$ is ruled by the following first–order differential equation:

$$\dot{z}(t) = h(t)\frac{A(t)\dot{x}(t) - \nu(t)\left(\beta|\dot{x}(t)||z(t)|^{n-1}z(t) + \gamma\dot{x}(t)|z(t)|^n\right)}{\eta(t)} \tag{2-9}$$

with the initial condition $z(0) = 0$. Here, the parameters $\beta$, $\gamma$ and $n$ control the shape of the loops, the parameters $\nu(t)$, $\eta(t)$ and $h(t)$ are associated respectively to the strength, stiffness and pinching degradation effects, and they are defined as linearly increasing functions of the absorbed hysteretic energy $\varepsilon(t)$ as:

$$\nu(t) := \nu_0 + \delta_\nu\varepsilon(t) \qquad A(t) := A_0 - \delta_A\varepsilon(t) \qquad \eta(t) := \eta_0 + \delta_\eta\varepsilon(t)$$

where $\delta_\nu$, $\delta_A$ and $\delta_\eta$ are constant parameters which determine the rate of degradation (in case we do not want to model degradation, we simply set those parameters to zero); here the absorbed hysteretic energy $\varepsilon(t)$ represents the energy dissipated by the hysteretic system and is defined as the area under the mass normalized hysteretic restoring force $F^h(z(t)) := (1-\alpha)k_i z(t)$ vs the total displacement; therefore, the absorbed hysteretic energy per unit mass, can be quantified as [7, 38]

$$\varepsilon(t) := \int_{x(0)}^{x(t)} \frac{F^h(x)}{m}\mathrm{d}x = (1-\alpha)\frac{k_i}{m}\int_0^t z(\tau)\dot{x}(\tau)\mathrm{d}\tau$$

that is,

$$\varepsilon(t) = (1-\alpha)\omega_0^2 \int_0^t z(\tau)\dot{x}(\tau)\mathrm{d}\tau. \tag{2-10}$$

Finally, the $h(t)$ term in Eq. (2-9) represents the pinching function and is defined as:

$$h(t) := 1 - \varsigma_1(t)\exp\left(-\frac{\Big(z(t)\,\mathrm{sign}(\dot{x}(t)) - qz_u\Big)^2}{\Big(\varsigma_2(t)\Big)^2}\right)$$

where $z_u$ is the ultimate value of $z$,

$$z_u = \sqrt[n]{\frac{1}{\nu(t)(\beta+\gamma)}}$$

and

$$\varsigma_1(t) := \Big(1-\exp\left(-p\varepsilon(t)\right)\Big)\varsigma_0 \qquad\qquad \varsigma_2(t) := \Big(\psi_0 + \delta_\psi\varepsilon(t)\Big)\Big(\lambda + \varsigma_1(t)\Big).$$

Here the constants $p$, $q$, $\varsigma_0$, $\psi_0$, $\delta_\psi$ and $\lambda$ define the form of the pinching. Finally, when $h(t) := 1$, pinching is not included in the model.

# Part II.

# Evolutionary algorithms

# 3. The optimization problem

## 3.1. Introduction

Many practical applications are described by mathematical models which involve parameters that must be tuned in order to get the desired performance and/or results. Optimization is the process of finding, among a set of alternatives, the best set of parameters and/or solutions of such a model.
In order to find the best set of variables that optimizes the model, the problem at hand must be formulated as an optimization problem, so a function (or set of functions) depending on the parameters of the original model will be used to summarize the final result. Such function(s) will be optimized (i.e., minimized or maximized depending on the situation) by tuning the model parameters; the optimal result will be chosen as the solution to the optimization problem.
A huge variety of problems from several disciplines can be viewed as optimization problems, such as [69]: allocation, planning, control, approximation, estimation, games, etc.

## 3.2. Definitions and problem statement

A brief description of the elements involved in the process of solving optimization problems is given below.

- **Decision variables** ($\boldsymbol{x}$): Is the set of variables (parameters) that act as input of the optimization problem. Generally $\boldsymbol{x} \in \mathbb{R}^q$ ($q$ is the number of parameters).

- **Model** ($\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{u})$): Set of equations used to represent mathematically the physical system under study.

- **Objective function** ($\boldsymbol{f}(\boldsymbol{x})$): Also known as *cost function*. Is the function (or set of functions) to be optimized. The objective function depends on the decision variables $\boldsymbol{x}$ and the output of the model $\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x}, \boldsymbol{u})$. There will be $m$ objective functions, i.e., $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T$.

- **Restrictions:** Also known as *constraints*. These are equations or inequalities used to restrict values of $\boldsymbol{x}$ in order to determine whether the selected parameters of the model

are feasible or not. Generally, constraints are determined by the physical properties of the system under study.

- **Decision space** ($\mathcal{D} \subseteq \mathbb{R}^q$): Also known as parameter space. Is the region of space bounded by the lower and upper boundaries of $\boldsymbol{x}$, and by the constraints defined in the optimization problem (See Fig. **3-1**).
- **Objective space** ($\mathcal{Z} \subseteq \mathbb{R}^m$): Also known as search space. Is the region where all elements of $\mathcal{D}$ are mapped through the function $\boldsymbol{f}$ (See Fig. **3-1**).

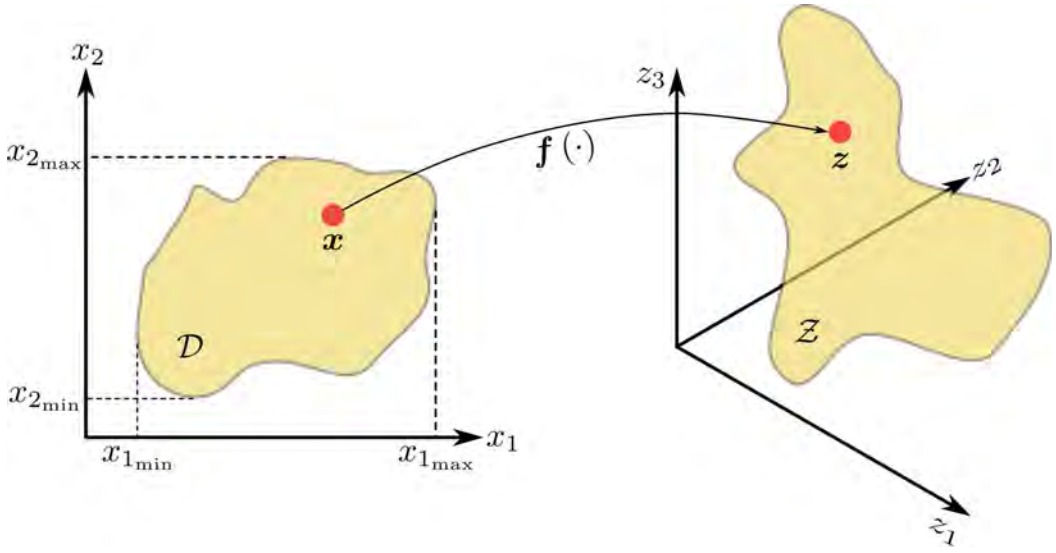



**Figure 3-1**.: Representation of decision space ($\mathcal{D}$) and objective space ($\mathcal{Z}$)

With the definitions given above, an optimization problem can be defined as:

$$\begin{aligned} \text{Minimize/Maximize:} \quad & \boldsymbol{f}(\boldsymbol{x}) \\ \text{subject to (s.t.)} \quad & \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{0} \\ & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \\ \text{for} \quad & \boldsymbol{x} \in \mathcal{D}; \end{aligned} \tag{3-1}$$

here $\boldsymbol{g}(\boldsymbol{x})$ is the set of *inequality constraints* and $\boldsymbol{h}(\boldsymbol{x})$ is the set of *equality constraints*. There will be up to $j$ inequality constraints and up to $k$ equality constraints. A set of decision variables that do not satisfy all $j + k$ constraints is considered an *infeasible point*, and consequently is not a solution of the optimization problem.
As it can be seen, the objective of the optimization problem is to find a point $\boldsymbol{x}^*$ such that this set of parameters is a feasible point of $\mathcal{D}$ and it minimizes (or maximizes) the objective function $\boldsymbol{f}(\boldsymbol{x})$. Note that the optimal point is represented by $\boldsymbol{x}^*$ instead of $\boldsymbol{f}(\boldsymbol{x}^*)$ because some problems present several optimal points, so a unique point $\boldsymbol{x}^*$ is difficult to characterize, whereas that the value representing the optimal point is unique.

## 3.3. Solving optimization problems

In optimization several methodologies have been proposed in order to compute the global optimum of functions. Basically, there are two approaches to find the optimum of functions in the search space: *gradient–based* methods and *population–based* methods.
The first approximation, as its name indicates, uses gradient–based search procedures in order to reach the final goal (the global optimum); examples of this type of approximation are the *steepest descent method*, *Newton method* and the *conjugate gradient method*. Such techniques are efficient in low dimensions if there is only a global minimum; in the other hand, they have some disadvantages, mainly the computational burden required to compute derivatives and the tendency to get stuck in local minima if good starting points are not provided.
The second approach is more flexible and easier to implement in computer programs, because algorithms based on that concept do not use derivatives (which are time consuming), instead, they perform a search in the whole space looking for the optimum using statistical operations to define the search direction. Examples of this approach are *evolutionary Algorithms*, *simulated annealing* and *particle swarm techniques*.
In the next chapter an in depth study of evolutionary algorithms will be presented; it will be seen how this heuristic–based techniques are capable of handle a huge variety of optimization problems, independent if they are uni–modal problems (one global optimum) or multi–modal problems (more than one global optimum).

# 4. Evolutionary algorithms

The main aim of this Chapter is to give a brief introduction to *evolutionary algorithms* (EA), which are a set of population–based optimization methods whose principles rely on natural evolution. This chapter follows the next plan: Section 4.1 gives the underlying theory of evolutionary algorithms. Section 4.2 introduces *genetic algorithms* (GAs) which are one of the mainstream algorithms derived from EAs. Evolution strategies (ES) are introduced in section 4.3, which are very popular in the field of optimization based on populations. Some examples that illustrate the applicability to optimization problems of EAs are presented in Section 4.4. Final remarks about these methodologies are given in Section 4.5. For a more comprehensive study on the topics presented in this chapter, the reader is referred to [33] and [8].

## 4.1. Introduction

Evolutionary algorithms (EAs) mimic the behaviour of populations in a natural environment by means of operators that simulate interactions between members of the population and the surrounding environment, and within the members of the population itself. Basically, an EA is an iterative method where a set of samples (population), generated in the *decision* space, are evolved towards better regions through *evolutionary operators* that imitate the natural process of evolution. These methods are considered to be *robust* in the sense that they are stable numerically and the results are not corrupted by inputs that belong or not to the decision space; also EAs are very suitable when little information about the problem at hand is available [33].
There are three major evolutionary operators involved in EAs:

**Reproduction:** Just as in nature, members of the population interact with each other, giving the chance to generate children with the same characteristics (information) of the parents.

**Mutation:** Some individuals suffer alterations on its genetic program, so individuals with new characteristics may appear in the population.

**Selection:** Individuals with the best *fitness values* are kept in the population. Members with the worst fitness values face extinction, and consequently, are deleted from the popu-

lation. Here, *fitness value* means a measure used to evaluate the success of individuals in reaching the global optimum.

### 4.1.1. Mathematical overview of EAs

Given a population of $\mu$ individuals at time $t$, $\boldsymbol{P}(t) = \{\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \ldots, \boldsymbol{x}_\mu(t)\}$, where $\boldsymbol{x}_i \in \mathcal{D} \subseteq \mathbb{R}^q$, $1 \leq i \leq \mu$ ($\mu$ is the parent population size). Let $\boldsymbol{f}(\boldsymbol{x}(t))$ be the fitness function, given by:

$$\begin{aligned} \boldsymbol{f} : \mathcal{D} &\to \mathcal{Z} \\ \boldsymbol{x}(t) &\mapsto \boldsymbol{f}\left(\boldsymbol{x}(t)\right) = \boldsymbol{z}, \end{aligned}$$

where $\mathcal{Z}$ is the so-called *objective space*, and $\boldsymbol{z} \in \mathcal{Z} \subseteq \mathbb{R}^m$ is an element in this space. The fitness of the whole population can be represented as:

$$\boldsymbol{F}(t) = \{\boldsymbol{f}(\boldsymbol{x}_1(t)), \boldsymbol{f}(\boldsymbol{x}_2(t)), \ldots, \boldsymbol{f}(\boldsymbol{x}_\mu(t))\}.$$

Now consider the operators $\Theta_r$, $\Theta_m$, $\Theta_s$, which are the *recombination (crossover)*, *mutation* and *selection* operators respectively, defined as:

$$\begin{aligned} \Theta_r : \mathcal{D} &\to \mathcal{D}' \subseteq \mathbb{R}^q \\ \boldsymbol{P}(t) &\mapsto \Theta_r\left(\boldsymbol{P}(t)\right) = \boldsymbol{P}'(t), \\ \Theta_m : \mathcal{D}' &\to \mathcal{D}'' \subseteq \mathbb{R}^q \\ \boldsymbol{P}'(t) &\mapsto \Theta_m\left(\boldsymbol{P}'(t)\right) = \boldsymbol{P}''(t), \\ \Theta_s : \mathcal{D}'' &\to \mathcal{D} \subseteq \mathbb{R}^q \\ \boldsymbol{Q}(t) &\mapsto \Theta_s\left(\boldsymbol{Q}(t)\right) = \boldsymbol{P}(t+1), \end{aligned}$$

where $\boldsymbol{P}'(t) = \{\boldsymbol{x}'_1(t), \boldsymbol{x}'_2(t), \ldots, \boldsymbol{x}'_\theta(t)\}$ is the population at time $t$ after recombination and $\boldsymbol{P}''(t) = \{\boldsymbol{x}''_1(t), \boldsymbol{x}''_2(t), \ldots, \boldsymbol{x}''_\theta(t)\}$ is the population at time $t$ after mutation. Note that $\boldsymbol{P}'(t)$ and $\boldsymbol{P}''(t)$ have $\theta$ elements (so $\theta$ stands for the *offspring population size*). The operator $\Theta_s$ changes its domain, depending on the inclusion or not of *elitism*, which is a mechanism used to preserve better solutions throughout the whole evolution process. Basically, elitism procedure ensures that the best individuals are kept, no matter if they belong to the parent population or the offspring population. In such a case, $\boldsymbol{Q}(t)$ can have $\theta$ elements (no elitism) or $\mu + \theta$ elements (elitism), and the elements from $\boldsymbol{Q}(t)$ are mapped to the parameter space $\mathcal{D}$. The whole process of taking elements from $\mathcal{D}$ and subject them to the operators $\Theta_r$, $\Theta_m$ and $\Theta_s$ is repeated until a *termination criterion* is satisfied. Generally, one of the following criteria is used to finish the process:

- Before starting the process, the maximum number of *generations* is provided.

- There are functions with many local minima surrounding the global minimum; even EAs can stagnate in one local minima, so another criterion is to define a *minimal improvement* of the objective value: Given a predefined tolerance $tol$ ($tol \ll 1$); if $|\boldsymbol{f}_k(\boldsymbol{x}(t)) - \boldsymbol{f}_{k+1}(\boldsymbol{x}(t))| \leq tol$, where $\boldsymbol{f}_k$ and $\boldsymbol{f}_{k+1}$ represent two successive iterations, then the algorithm is finished.

- If the optimum value of the objective value is known, say $\boldsymbol{f}(\boldsymbol{x}_{\text{opt}}(t))$. Given a point $\boldsymbol{x}(t) \in \mathcal{D}$ such that $||\boldsymbol{f}(\boldsymbol{x}_{\text{opt}}(t)) - \boldsymbol{f}(\boldsymbol{x}(t))|| \leq tol$, where $tol$ is a predefined tolerance ($tol \ll 1$), then $\boldsymbol{x}(t)$ can be considered a good approximation to the optimum, so the algorithm can finish here.

**Procedure 4.1** Evolutionary Algorithm (EA)

$t = 0$
initialize $\boldsymbol{P}(0) \rightarrow \{\boldsymbol{x}_1(0), \ldots, \boldsymbol{x}_\mu(0)\}$, $\boldsymbol{x}_i(0) \in \mathcal{D} \subseteq \mathbb{R}^q$, $1 \leq i \leq \mu$
evaluate $\boldsymbol{P}(0) \rightarrow \boldsymbol{F}(0) = \{\boldsymbol{f}(\boldsymbol{x}_1(0)), \ldots, \boldsymbol{f}(\boldsymbol{x}_\mu(0))\}$, $\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{Z} \subseteq \mathbb{R}^m$
**repeat**
  recombine $\boldsymbol{P}(t) \rightarrow \boldsymbol{P}'(t) = \Theta_r(\boldsymbol{P}(t))$
  mutate $\boldsymbol{P}'(t) \rightarrow \boldsymbol{P}''(t) = \Theta_m(\boldsymbol{P}'(t))$
  **if not** elitism **then**
    $\boldsymbol{Q}(t) = \boldsymbol{P}''(t)$
    evaluate $\boldsymbol{Q}(t) \rightarrow \boldsymbol{F}(t) = \{\boldsymbol{f}(\boldsymbol{x}''_1(t)), \ldots, \boldsymbol{f}(\boldsymbol{x}''_\theta(t))\}$
  **else if** elitism **then**
    $\boldsymbol{Q}(t) = \boldsymbol{P}''(t) \bigcup \boldsymbol{P}(t)$
    evaluate $\boldsymbol{Q}(t) \rightarrow \boldsymbol{F}(t) = \left\{\boldsymbol{f}(\boldsymbol{x}''_1(t)), \ldots, \boldsymbol{f}\left(\boldsymbol{x}''_{\mu+\theta}(t)\right)\right\}$
  **end if**
  select $\boldsymbol{P}(t+1) \rightarrow \boldsymbol{P}(t+1) = \Theta_s(\boldsymbol{Q}(t))$
  $t \rightarrow t+1$
**until** termination criterion is met

Algorithms derived from EAs have essentially the same structure described in Procedure 4.1. They differ each other in the way and order they apply the operators $\Theta_r$, $\Theta_m$ and $\Theta_s$. Next sections are intended to give a brief description and differences among genetic algorithms and evolution strategies, which are the most important techniques based on the theory presented so far.

## 4.2. Genetic algorithms (GAs)

Genetic algorithms are probably the most known EAs used in derivative–free optimization. They have shown their applicability in many fields such industry, engineering, science, among

others. GAs were proposed by Holland [49, 50] in order to resolve problems arising in practical applications by means of algorithms that simulate biological systems [8].
Initially, GAs were proposed to work in a *binary–code* fashion, so the algorithm could imitate the genetic encoding of individuals, but there are variants that work with *real–parameters* instead, because some problems are easier to handle in such a way, avoiding the codification step needed in the classical GAs.

### 4.2.1. Binary–coded GAs

Binary–coded GAs represent each individual in the population by *binary strings* of fixed length $l$. There are different ways to carry out the codification of a population, the most popular binary encoding systems used by GAs are *plain binary representation* (base two representation) and *Gray code* system [8].
As said before, all optimization techniques derived from EAs follow a similar process described in Procedure 4.1. So the first step states that an initial population of guesses $\boldsymbol{P}(0)$ is created (usually these individuals are randomly generated and spread throughout the parameter space $\mathcal{D}$), $\boldsymbol{x}^l \leq \boldsymbol{x}_i(t) \leq \boldsymbol{x}^u$, $1 \leq i \leq \mu$, where $\boldsymbol{x}^l$ and $\boldsymbol{x}^u$ are the lower and upper boundary of the population $\boldsymbol{P}(t)$ respectively, and define the feasible decision space. Then, according to binary–coded GAs, the population is encoded using any encoding system.
After the initial population has been created, their *fitness value* are computed using a *decoding* process and evaluating the fitness function of each member; the vast majority of cases the fitness function is equal to the objective function value. The following step is to apply the evolutionary operators, which in GAs are natural selection, recombination (crossover) and mutation.

#### Natural selection

Now that the fitness value of each individual has been computed, a selection process is performed in order to select the best members to form the *mating pool*, which will contain the eligible members for reproduction and mutation. There are several techniques used to carry out the selection, the two most important are the following:

- **Tournament selection:** Two members of the population, chosen randomly, are compared with respect to their fitness values, the best solutions are stored in the mating pool. Each member can compete in two tournaments, so each member will have at most none, one or two copies in the mating pool. This process is repeated until each member has played two tournaments, and consequently, all the population has been used in the process. The mating pool will have $\mu$ individuals.

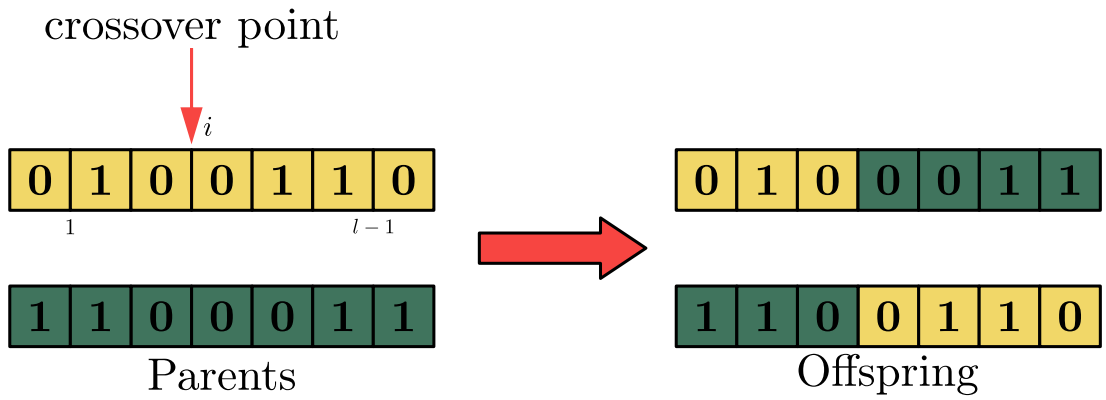
- **Roulette wheel:** Each individual is mapped to contiguous segments of the real number line $[0, 1)$. Given that some members will have better fitness values than others, there will be segments of different sizes according to such values. Then, a random

number $\chi \in [0, 1)$ is generated, so the individual whose segment contains $\chi$ is stored in the mating pool. The process is repeated $\mu$ times.

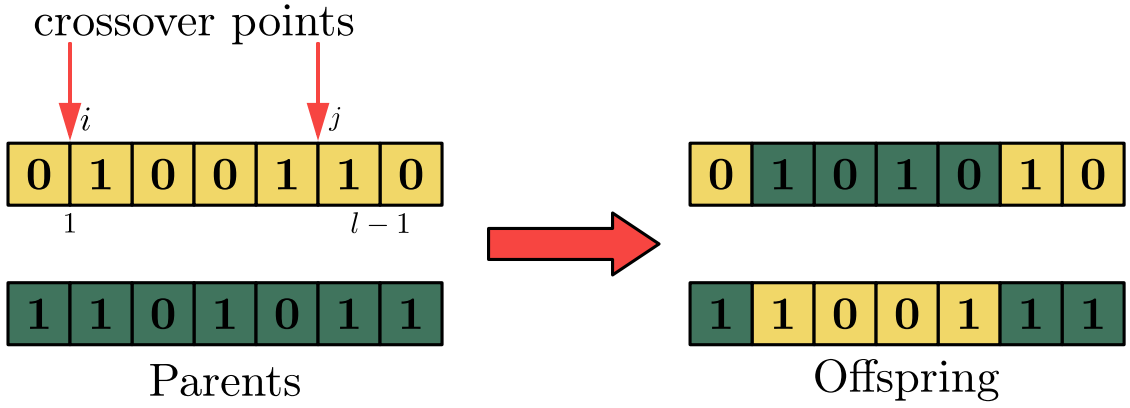
### Recombination or crossover

This is the main operator in GAs, because the search characteristic of GAs is provided by this operation. The idea is to select randomly pairs of individuals (parents) from the mating pool and interchange information between them in order to create new members (offspring) with combined binary information from the parents. It is expected that the new individuals retain good characteristics from its parents, and therefore, some children will be able to outperform their predecessors [56].

The most popular and basic crossover operator is the *one–point crossover* (Fig. **4-1**(a)). Remember that each individual has a binary string of length $l$, so the one–point crossover operator chooses a point between 1 and $l-1$ and interchange all the binary information to the right of such point between the two members involved in the process. One drawback of using this crossover operator is its *positional bias* [8], so one solution is to increase the number of crossover points as shown in Fig. **4-1**(b).




(a) One–point crossover




(b) Two–point crossover

**Figure 4-1**.: Crossover in binary–coded GAs

A *crossover probability* $p_c$ ($0 \leq p_c \leq 1$) is defined in order to perform this operation. $100p_c\%$ of the population will be used in the crossover operation, the remaining individuals are passed to the new population as their own offspring.

#### Mutation

Now that the offspring population has been generated, some members of the offspring population may suffer mutations. The process of mutation is basically a bit wise operator which introduces new information to the population just by flipping bits with a probability $p_m$ (mutation probability). Every time a mutation will be performed, a random number $\chi \in [0, 1]$ is generated, if $\chi \geq p_m$, then the mutation is not performed, conversely, if $\chi < p_m$, then the mutation is successfully applied to one member of the offspring population. Fig. **4-2** shows an example of mutation.

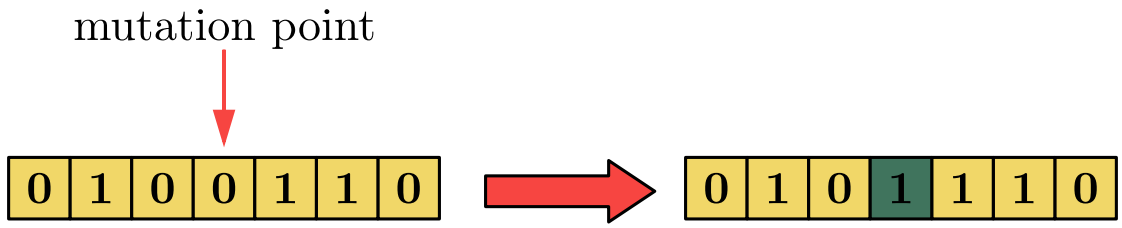



**Figure 4-2**.: Mutation in binary–coded GAs

It is worth noting that $p_m$ is usually a small number, because, as in nature, mutations seldom occur. Mutation is applied in order to keep diversity among the population, and sometimes, it is useful when some local minima are surrounding a global minimum, because the mutated individual can escape from local minima and maybe reach the global minimum, and consequently, attract the rest of the population to such point.
The three operators described above are applied iteratively until a stop criterion is met by the algorithm. The main idea behind them is to try duplicating better solutions each generation, and consequently the last generations will give emphasis to those regions where individuals with the best fitness values are located.

### 4.2.2. Real–parameter GAs

Real–parameter GAs are used mainly in applications where the search space is a set of continuous points. Binary–coded GAs can be implemented in such cases, but some problems arise when using them in continuous search spaces, such as:

- **Precision:** The precision required in binary–coded GAs to represent individuals within a population is a function of the string length $l$: the higher the required precision, the longer the bit string. When $l$ is increased to reach certain precision, the computational

requirements are higher, because the algorithm has to store $l$ integers instead of a single floating–point that such string is representing.

- **Hamming cliffs:** When the current population of encoded individuals begins to converge to a global optimum, is expected that the optimization problem becomes a gradual search towards that point. For that reason, the next generation is likely to appear in a neighborhood of the actual population. But one problem that binary–coded GAs face is the difficulty in the transition between points in a neighborhood.

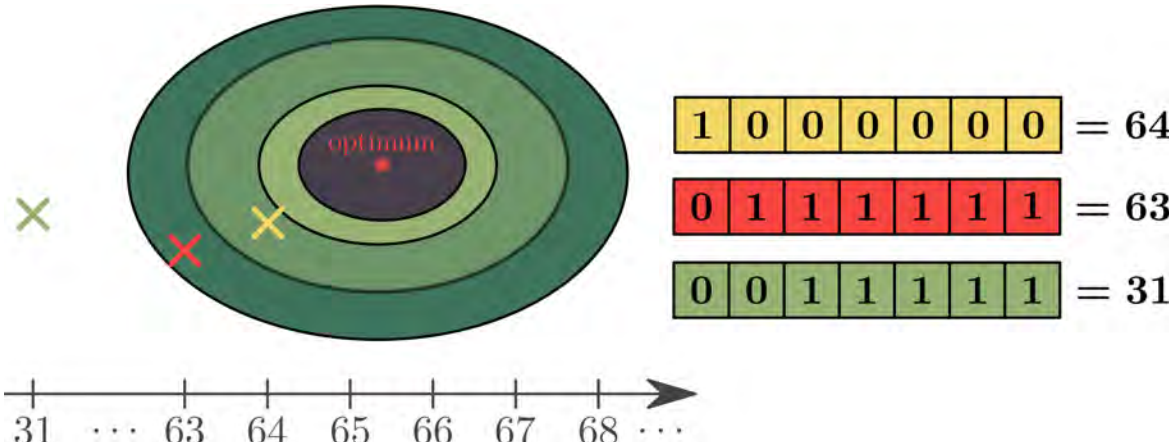



**Figure 4-3**.: Hamming Cliffs in binary–coded GAs

Consider the case shown in Fig. **4-3**, two points in a neighborhood of the optimum (63 and 64) are represented by bit strings completely different; when evolutionary operators modify one of those points, the new member is likely to become worse than the original one, because in order to make the transition from the bit string representing 63 to the bit string representing 64, all the information in one of the strings has to be modified, and usual mutation and crossover operators cannot make such change. This phenomenon is known as *Hamming cliffs*, a huge difference between two bit strings that belong to points in a neighborhood.

Solutions have been proposed in order to tackle these problems, for instance, *gray encoding system* is used to solve the problem related with Hamming cliffs, but such encoding system introduces nonlinearities between the bit string and the decoded element [16], moreover, the problem of precision is not solved yet. For these reasons, real–parameter GAs are an attractive option when dealing with continuous problems, not only because of their faster performance, but also because of their simplicity against the binary–coded GAs which have to add steps of codification/decodification to carry out the evolutionary process of the population.

Just as any EA, an initial population has to be created in the parameter space $\mathcal{D}$ (usually random generation of samples), $\boldsymbol{x}^l(t) \leq \boldsymbol{x}_i(t) \leq \boldsymbol{x}^u(t)$, $1 \leq i \leq \mu$. Then, the fitness value of

each sample is computed. Now, evolutionary operators are applied, however, given the new structure of the individuals, such operators have to be modified accordingly.

### Natural selection

Unlike binary–coded GAs, the population in real–parameter GAs do not need a codification, so the real values of the variables can be used directly to evaluate the fitness of each individual. Any of the selection processes described in binary–coded GAs can be used to determine the mating pool, the procedures do not suffer modifications at all.

### Crossover

There are several crossover operators in real–parameter GAs, the most popular are given in the following list:

- Linear crossover.
- Blend crossover.
- Simplex crossover.
- Fuzzy recombination operator.
- Unimodal normally distributed crossover.
- Simulated binary crossover.

Next, a brief description of the simulated binary crossover operator is given. For more information about the other crossover operators, the reader is referred to [33].
Basically, the simulated binary crossover procedure generates two children, namely $\boldsymbol{x}'_i(t)$, $\boldsymbol{x}'_j(t) \in \boldsymbol{P}'(t)$, from two parents selected randomly from the mating pool $\boldsymbol{x}_i(t), \boldsymbol{x}_j(t) \in \boldsymbol{P}(t)$. A *crossover distribution index* $\eta_c$ has to be defined, which is a positive real number ($\eta_c \in \mathbb{R}^+$) and whose value gives the possibility to generate children close to their parents (high values of $\eta_c$) or generate a distant offspring (small value of $\eta_c$). Then, a *spread factor* $\beta_{qi}$ is computed with the following expression:

$$\beta_{qi} = \begin{cases} (2\chi)^{\frac{1}{\eta_c+1}}, & \text{if } \chi \leq 0.5 \\ \left(\frac{1}{2(1-\chi)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases}$$

where $\chi$ is a random number ($\chi \in [0, 1)$). Finally, the offspring are calculated as follows:

$$\boldsymbol{x}'_i(t) = 0.5\left[(1+\beta_{qi})\,\boldsymbol{x}_i(t) + (1-\beta_{qi})\,\boldsymbol{x}_j(t)\right],$$
$$\boldsymbol{x}'_j(t) = 0.5\left[(1-\beta_{qi})\,\boldsymbol{x}_i(t) + (1+\beta_{qi})\,\boldsymbol{x}_j(t)\right].$$

Just as in the binary–coded case, a crossover probability $p_c$ ($0 \leq p_c \leq 1$) is defined in order to perform this operation, $100p_c\%$ of the population will be used in the crossover operation, the remaining individuals are copied to the new population.

### Mutation

There are various types of mutation operators involved with real–parameter GAs, the difference against the crossover operator, is that in mutation, just one individual is involved in the process, whereas that in crossover, at least two individuals are needed to perform this evolutionary stage. Some of the most common real–parameter mutation operators are:

- Non–uniform mutation.
- Normally distributed mutation.
- Polynomial mutation.

The polynomial mutation is described next. For an explanation about the other mutation schemes, the reader is referred to [33].
Just like the simulated binary crossover, a *mutation distribution index* $\eta_m$ is defined ($\eta_m \in \mathbb{R}^+$). $\eta_m$ plays a similar role to $\eta_c$, so a large value of $\eta_m$ produces a small perturbation in the individual being mutated, while a small value of $\eta_m$ generates a considerable perturbation of the individual. A parameter $\bar{\delta}$ is computed from the following expression:

$$\bar{\delta} = \begin{cases} (2\chi)^{\frac{1}{\eta_m+1}} - 1, & \text{if } \chi < 0.5 \\ 1 - [2\,(1-\chi)]^{\frac{1}{\eta_m+1}}, & \text{if } \chi \geq 0.5 \end{cases}$$

where $\chi$ is a random number ($\chi \in [0, 1)$). Then the individual $\boldsymbol{x}'_i(t) \in \boldsymbol{P}'(t)$ is mutated using the following formula:

$$\boldsymbol{x}''_i(t) = \boldsymbol{x}'_i(t) + \left(\boldsymbol{x}^u - \boldsymbol{x}^l\right)\bar{\delta}.$$

Not all the members of the population have to be mutated, just a few, according to a mutation probability ($p_m$). The decision of apply or not mutation to the population is given by the same conditions as binary–coded GAs.
The evolutionary process described by the three operators (selection, crossover, mutation) is applied iteratively until a stop criterion is met by the algorithm.

## 4.3. Evolution strategies (ES)

Evolution strategies (ES) are EAs used in optimization problems related with real–parameter values. They were proposed by Rechenberg [81] and Schwefel [84] to solve shape optimization problems. ES employ mechanisms that are *self–adaptive* according to the topology of the search space, and this feature is what sets it apart from GAs.
ES have been modified since its inception because the first algorithms were simple and had poor performance against other methods, such as GAs for instance. Nowadays, two ES are basically identified: $(\mu, \theta)$–ES and $(\mu + \theta)$–ES, where $\mu$ stands for the parent population

size and $\theta$ for the offspring population size. Both methods have the same procedure, the only difference lies in the selection stage, where in the $(\mu, \theta)$–ES the parent population is replaced by the offspring population, while in the $(\mu + \theta)$–ES both populations, parents and offspring, are taken into account to create the next generation $\boldsymbol{P}(t+1)$, this is the reason why these methods are known as *selection schemes*, because they just modify this operator in the original ES. It is worth noting that in the $(\mu, \theta)$–ES, $\theta \geq \mu$ in order to generate at least $\mu$ individuals for the next generation; in the $(\mu + \theta)$–ES this is not a requirement, so $\theta$ can be greater or less than $\mu$.

Just as any EA, an initial population $\boldsymbol{P}(0)$ has to be generated, but ES have a peculiar feature in this step, they assign *strategy parameters* to each individual $\boldsymbol{x}(t)$ in order to improve the search capabilities of the algorithm and determine the best step size that each individual has to take in the search space towards the optimum.

In ES, each individual is represented by a triplet $(\boldsymbol{x}(t), \boldsymbol{\sigma}, \boldsymbol{A})$, where $\boldsymbol{x}(t) \in \mathbb{R}^q$ is the individual itself, $\boldsymbol{\sigma}$ is a vector whose elements are the *standard deviations* $\sigma_i$ of the elements $x_i(t)$, $1 \leq i \leq q$, and $\boldsymbol{A}$ is a matrix whose elements are *rotation angles* which determine the best search direction of each individual.

Generally, the number of standard deviations $(n_\sigma)$ is set to $q$, but this is not mandatory, so $1 \leq n_\sigma \leq q$ is allowed. If $n_\sigma < q$, then the standard deviations $\sigma_1, \ldots, \sigma_{n_\sigma - 1}$ are coupled with the variables $x_1, \ldots, x_{n_\sigma - 1}$, and $\sigma_{n_\sigma}$ is coupled with the remaining variables $x_{n_\sigma}, \ldots, x_q$.

In ES, an $q \times q$ symmetrical matrix $\boldsymbol{\Sigma}$ (the *covariance matrix*) is defined for each individual instead of a vector $\boldsymbol{\sigma}$, the diagonal elements of $\boldsymbol{\Sigma}$ are the *variances* $(\sigma_i^2,\ 1 \leq i \leq q)$, and the off-diagonal elements are the *covariances* $(c_{ij})$.

The number of rotation angles $(n_\alpha)$ depends on $n_\sigma$ and $q$, and is given by the following expression:

$$n_\alpha = \frac{(2q - n_\sigma)\,(n_\sigma - 1)}{2},$$

but $n_\alpha$ can also be set to zero if this strategy parameter is unwanted. The matrix $\boldsymbol{A}$ is an $q \times q$ matrix with the following layout:

$$\boldsymbol{A} = \begin{bmatrix} 0 & \alpha_{12} & \alpha_{13} & \alpha_{14} & \cdots & \alpha_{1,q-1} & \alpha_{1,q} \\ 0 & 0 & \alpha_{23} & \alpha_{24} & \cdots & \alpha_{2,q-1} & \alpha_{2,q} \\ 0 & 0 & 0 & \alpha_{34} & \cdots & \alpha_{3,q-1} & \alpha_{3,q} \\ 0 & 0 & 0 & 0 & \cdots & \alpha_{4,q-1} & \alpha_{4,q} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_{q-1,q} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{4-1}$$

where each rotation angle $\alpha_{ij}$ $(1 \leq i \leq q-1,\ i+1 \leq j \leq q)$ is computed from the covariance

matrix $\boldsymbol{\Sigma}$ as follows:

$$\alpha_{ij} = \frac{1}{2}\arctan\left(\frac{2c_{ij}}{\sigma_i^2 - \sigma_j^2}\right).$$

After each individual $\boldsymbol{x}(t)$ has been initialized with its respective strategy parameters $(\boldsymbol{\Sigma}, \boldsymbol{A})$, it is time to assign a fitness value which, as in GAs, is computed from the fitness function. Conventionally, the fitness function is the same objective function. Then, evolutionary operators (Recombination, mutation and selection) are applied so new members with better fitness values are discovered in the decision space $\mathcal{D}$.

### 4.3.1. Recombination

There are basically two recombination schemes in ES: *discrete recombination* and *intermediate recombination*; each of these recombination mechanisms have two variants:

- **Sexual variant:** Two parents $(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t),\ 1 \leq i, j \leq \mu)$ are randomly selected from the parent population to generate a new member in the offspring population $(\boldsymbol{x}'_k(t),\ 1 \leq k \leq \theta)$. New individuals are generated from two parents.

- **Panmictic variant:** One parent $(\boldsymbol{x}_i(t))$ is randomly chosen from the parent population and is held fixed, while for each component $x_m(t) \in \boldsymbol{x}_i(t)$, $1 \leq m \leq q$, $1 \leq i \leq \mu$, a second parent $\boldsymbol{x}_j(t)$, $1 \leq j \leq \mu$, is chosen anew to create $x'_m(t) \in \boldsymbol{x}'_k(t)$, $1 \leq m \leq q$, $1 \leq k \leq \theta$. Unlike the sexual variant, a new member is generated from several parents (if $\mu > 1$).

A generalization of the intermediate recombination mechanism has been proposed by Schwefel in both variants (sexual and panmictic) introducing a weighting factor $\chi \in [0, 1]$. Next a brief description of each recombination scheme is given. Consider two parents $\boldsymbol{x}_i(t), \boldsymbol{x}_j(t) \in \boldsymbol{P}(t)$, $1 \leq i, j \leq \mu$, with components

$$\boldsymbol{x}_i(t) = (x_{1_i}(t), \ldots, x_{m_i}(t), \ldots, x_{q_i}(t))$$
$$\boldsymbol{x}_j(t) = \left(x_{1_j}(t), \ldots, x_{m_j}(t), \ldots, x_{q_j}(t)\right)$$

and a new member after recombination $\boldsymbol{x}'_k(t) \in \boldsymbol{P}'(t)$, $1 \leq k \leq \theta$, with components $\boldsymbol{x}'_k(t) = \left(x'_{1_k}, \ldots, x'_{m_k}, \ldots, x'_{q_k}\right)$.

- **No recombination:** This is the simplest mechanism to generate a new population $\boldsymbol{P}'(t)$, just a random selection is performed $\theta$ times from $\boldsymbol{P}(t)$, this means that $\boldsymbol{x}'_k(t) = \boldsymbol{x}_i(t)$.

- **Discrete recombination:** Two parents are randomly selected from $\boldsymbol{P}(t)$ and a new member $\boldsymbol{x}'_k(t)$ results from a random selection of each component between these parents, i.e., $x'_{m_k}(t) = x_{m_i}(t)$ or $x'_{m_k}(t) = x_{m_j}(t)$.

- **Panmictic discrete recombination:** One parent $\boldsymbol{x}_i(t)$ is randomly selected and held fixed. In order to generate each component $x'_{m_k}(t) \in \boldsymbol{x}'_k(t)$, another member $\boldsymbol{x}_j(t)$ is arbitrarily chosen and a discrete recombination is performed between $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$ just for one component of $\boldsymbol{x}'_k(t)$. To compute another component of $\boldsymbol{x}'_k(t)$, another member is randomly chosen from $\boldsymbol{P}(t)$, and another discrete recombination is performed between $\boldsymbol{x}_i(t)$ (the member that is kept fixed) and the new member chosen from $\boldsymbol{P}(t)$ (just for the $m$–th component). This process is repeated $q$ times; after that, a new member $\boldsymbol{x}_l(t) \in \boldsymbol{P}(t)$, $1 \leq l \leq \mu$ is randomly chosen and held fixed, and the previous process to generate a new member of $\boldsymbol{P}'(t)$ is repeated.

- **Intermediate recombination:** This mechanism computes the average of the components between two parents, i.e.:

  $$\boldsymbol{x}'_k(t) = \frac{1}{2}\left(\boldsymbol{x}_i(t) + \boldsymbol{x}_k(t)\right).$$

- **Panmictic intermediate recombination:** One parent $\boldsymbol{x}_i(t)$ is randomly selected and kept fixed. To generate each component $x'_{m_k}(t) \in \boldsymbol{x}'_k(t)$, other member $\boldsymbol{x}_j(t)$ is arbitrarily chosen and a intermediate recombination is performed between $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_j(t)$ just for one component of $\boldsymbol{x}'_k(t)$. To compute another component of $\boldsymbol{x}'_k(t)$, another member is randomly chosen from $\boldsymbol{P}(t)$, and another intermediate recombination is performed again between $\boldsymbol{x}_i(t)$ (the member that is held fixed) and the new member chosen from $\boldsymbol{P}(t)$ (just for the $m$–th component). This process is repeated $q$ times; after that, a new member $\boldsymbol{x}_l(t) \in \boldsymbol{P}(t)$, $1 \leq l \leq \mu$ is randomly chosen and held fixed, and the previous process to generate a new member of $\boldsymbol{P}'(t)$ is repeated.

- **Generalized intermediate recombination:** This recombination scheme is similar to the original intermediate recombination mechanism, but instead of computing the arithmetic mean between the two parents involved in the process, a random number $\chi \in [0, 1]$ is chosen, and the new member $\boldsymbol{x}'_k(t)$ is computed as follows:

  $$\boldsymbol{x}'_k(t) = \boldsymbol{x}_i(t) + \chi\left(\boldsymbol{x}_j(t) - \boldsymbol{x}_i(t)\right).$$

- **Panmictic generalized intermediate recombination:** This mechanism works under the same principle of the panmictic intermediate recombination, the difference lies in the usage of a random variable $\chi \in [0, 1]$ instead of the arithmetic mean to compute the components of each individual in the offspring population $\boldsymbol{P}'(t)$. A new random variable $\chi_m$ is generated every time a new component is calculated, i.e.:

  $$x'_{m_k}(t) = x_{m_i}(t) + \chi_m\left(x_{m_j}(t) - x_{m_i}(t)\right).$$

Taking a closer look to the recombination operators, it can be seen that the new members generated by these operators are limited by the parent population, i.e., the offspring population is always enclosed by the hypercube delimited by the parent population. For instance,

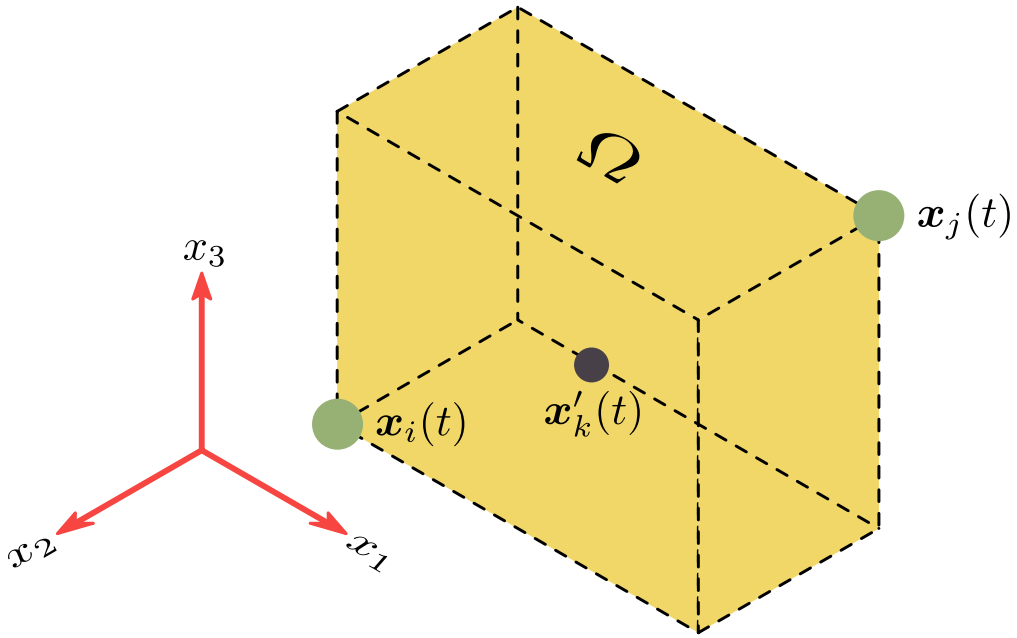



**Figure 4-4**.: Recombination scheme ES

Fig. 4-4 shows the case in a 3D space, two parents $\boldsymbol{x}_i(t), \boldsymbol{x}_j(t) \in \boldsymbol{P}(t)$ are recombined and a new individual $\boldsymbol{x}'_k(t)$ is generated inside the volume $\Omega$; it is impossible to generate a new member outside of that region, so recombination can be seen as a volume reduction of the parameter space [8].
It is worth noting that recombination not only applies to decision variables, but also to strategy parameters, so $\boldsymbol{\Sigma}$ and $\boldsymbol{A}$ are also recombined to accelerate the search process and improve self–adaptation of such parameters. According to Schwefel [85], a discrete recombination on decision variables and (panmictic) intermediate recombination of strategy parameters is a good selection of recombination schemes.

### 4.3.2. Mutation

Mutation takes a triplet $(\boldsymbol{x}'(t), \boldsymbol{\Sigma}', \boldsymbol{A}')$ and generates a new individual $(\boldsymbol{x}''(t), \boldsymbol{\Sigma}'', \boldsymbol{A}'')$ by means of the following operations:

$$\begin{aligned}
\boldsymbol{\Sigma}'' &= \boldsymbol{\Sigma}' \circ \exp\left(\tau' N\left(0,1\right) + \tau \boldsymbol{L}\right), \\
\boldsymbol{A}'' &= \boldsymbol{A}' + \beta \boldsymbol{U}, \\
\boldsymbol{x}''(t) &= \boldsymbol{x}'(t) + \boldsymbol{N}\left(\boldsymbol{\mathcal{O}}, \boldsymbol{\Sigma}'', \boldsymbol{A}''\right),
\end{aligned}$$

where $N\left(0,1\right)$ denotes a random number obtained from a normal distribution with mean zero and standard deviation one, $\boldsymbol{L}$ is an $q \times q$ random symmetric matrix whose elements are obtained from a normal distribution with mean zero and standard deviation one, $\circ$ represents the *Hadamard product* between two matrices (element–wise multiplication), $\boldsymbol{U}$ is an $q \times q$

upper triangular matrix whose main diagonal has zeros, i.e., it has the same layout as matrix $\boldsymbol{A}$ (see Eq. (4-1)). The constants $\tau'$, $\tau$ and $\beta$ are *learning rates* which are set to:

$$\tau \propto \frac{1}{\sqrt{2\sqrt{q}}},$$
$$\tau' \propto \frac{1}{\sqrt{2q}},$$
$$\beta \approx 0.0873,$$

where $\propto$ means *proportional to*; usually, the proportional constants for $\tau$ and $\tau'$ are set to one. $\boldsymbol{N}\left(\boldsymbol{\mathcal{O}}, \boldsymbol{\Sigma''}, \boldsymbol{A''}\right)$ is an $q \times 1$ normally distributed random vector with expectation zero, covariance matrix $\boldsymbol{\Sigma''}$ and rotation angles $\boldsymbol{A''}$, which is given by:

$$\boldsymbol{N}\left(\boldsymbol{\mathcal{O}}, \boldsymbol{\Sigma''}, \boldsymbol{A''}\right) = \boldsymbol{T}\left(\alpha''_{ij}\right) \boldsymbol{S}\left(\sigma''_{ii}\right) \boldsymbol{N}\left(0,1\right),$$

where $\boldsymbol{T}\left(\alpha''_{ij}\right)$ is the *Orthogonal rotation matrix* computed as:

$$\boldsymbol{T}\left(\alpha''_{ij}\right) = \prod_{i=1}^{q-1} \prod_{j=i+1}^{q} \boldsymbol{R}\left(\alpha''_{ij}\right),$$

which is the product of rotation matrices $\boldsymbol{R}(\alpha''_{ij})$ with ones in the main diagonal, entries $r_{ii} = r_{jj} = \cos(\alpha''_{ij})$, entries $r_{ij} = -r_{ji} = -\sin(\alpha''_{ij})$, and zeros elsewhere.
The matrix $\boldsymbol{S}\left(\sigma''_{ii}\right)$ is an $q \times q$ diagonal matrix whose elements are the square root of the diagonal elements of $\boldsymbol{\Sigma''}$.
$\boldsymbol{N}\left(0,1\right)$ is an $q \times 1$ vector whose elements are random numbers generated from a normal distribution with expectation zero and standard deviation one.
Given that the strategy parameters are modified by this evolutionary operator, it has to be ensured that the standard deviations do not become zero, and that the rotation angles do not leave the range $[-\pi, \pi]$. In the first case, a tolerance $\varepsilon_\sigma$ that forces all standard deviations to be greater than a certain value is predefined; in the second case, rotation angles whose absolute values are greater than $\pi$ are circularly mapped to the range $[-\pi, \pi]$, i.e.:

$$\begin{aligned} \text{if } \left|\alpha''_{ij}\right| > \pi \quad &\rightarrow \quad \alpha''_{ij} = \alpha''_{ij} - 2\pi \text{sign}\left(\alpha''_{ij}\right) \\ \text{if } \sigma''_{ii} < \varepsilon_\sigma \quad &\rightarrow \quad \sigma''_{ii} = \varepsilon_\sigma \end{aligned}$$

### 4.3.3. Selection

As said before, there are two selection schemes in ES: $(\mu, \theta)$–selection and $(\mu + \theta)$–selection. The difference between these two selection schemes lies in the population used to determine the next generation $\boldsymbol{P}(t+1)$.
The $(\mu, \theta)$–selection scheme evaluate the fitness of the offspring and determine $\boldsymbol{P}(t+1)$ from this population, the members with the best fitness values are passed to the next generation;

the actual parent population is erased, for that reason in this selection scheme the number of individuals in the offspring population must be greater or equal to the number of members in the parent population size ($\theta \geq \mu$), in order to generate at least $\mu$ new parents.
The $(\mu + \theta)$–selection scheme gathers both populations (parents and offspring) in one unique group, and the next generation is generated from this group. The new whole population is evaluated and the best members are passed directly to the next generation, it does not matter if the member is a parent or a descendant. Given that the parents are included in the selection stage, the number of members in the offspring population can be less, equal or greater than the number of parents.
With the new population $\boldsymbol{P}(t+1)$ already defined, the whole process is repeated (recombination, mutation, selection) until a convergence criterion is met.

## 4.4. Examples

To show the effectiveness of EAs for solving optimization problems, some test functions are given here and optimized using GAs and ES. The idea is to demonstrate the versatility and performance of these algorithms when handling functions with one or several minima. Two test functions (artificial landscapes) are optimized: *Himmelblau's function* (Fig. **4-5**(a)) and *Rastrigin's function* (Fig. **4-5**(b)), which are very popular objective functions used to evaluate the performance of optimization algorithms. Other test functions are given in Appendix A, all of them implemented in the routine `evolution_strategy.m` (See Appendix A).
Next a brief description about the parameters used in GAs and ES to optimize these functions, results and comparisons are given.

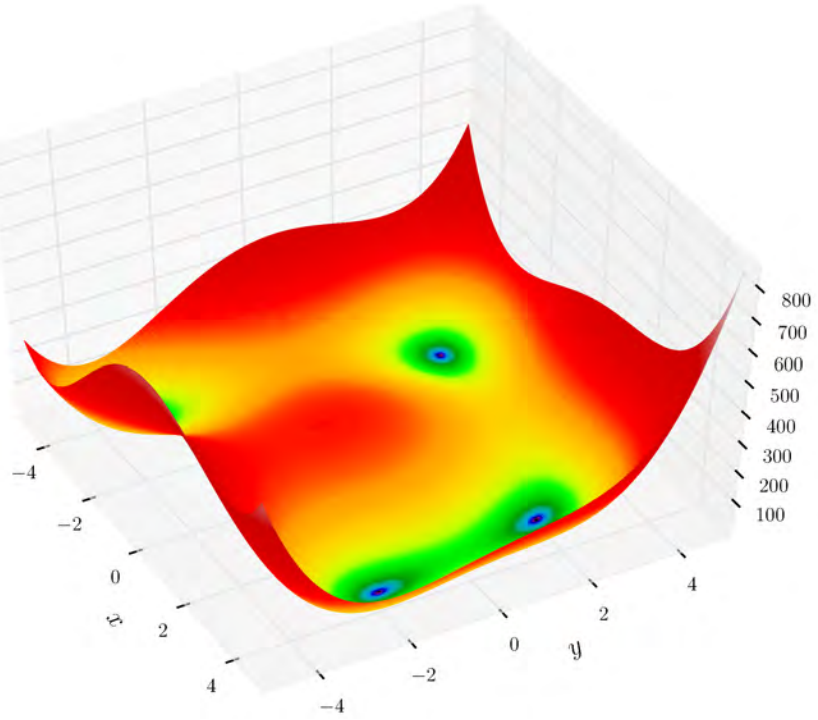
### 4.4.1. Genetic Algorithms (GA)

Real–parameter GAs were used here. An initial population $\boldsymbol{P}(0)$ with 100 parents was defined ($\mu = 100$), and the number of offspring individuals was also set to 100 ($\theta = 100$). The maximum number of iterations allowed (number of generations) was 100. Simulated binary crossover, with crossover distribution index $\eta_c = 15$, was used as mechanism to generate new members; and for mutation, the polynomial variant with mutation distribution index $\eta_m = 20$ was applied.

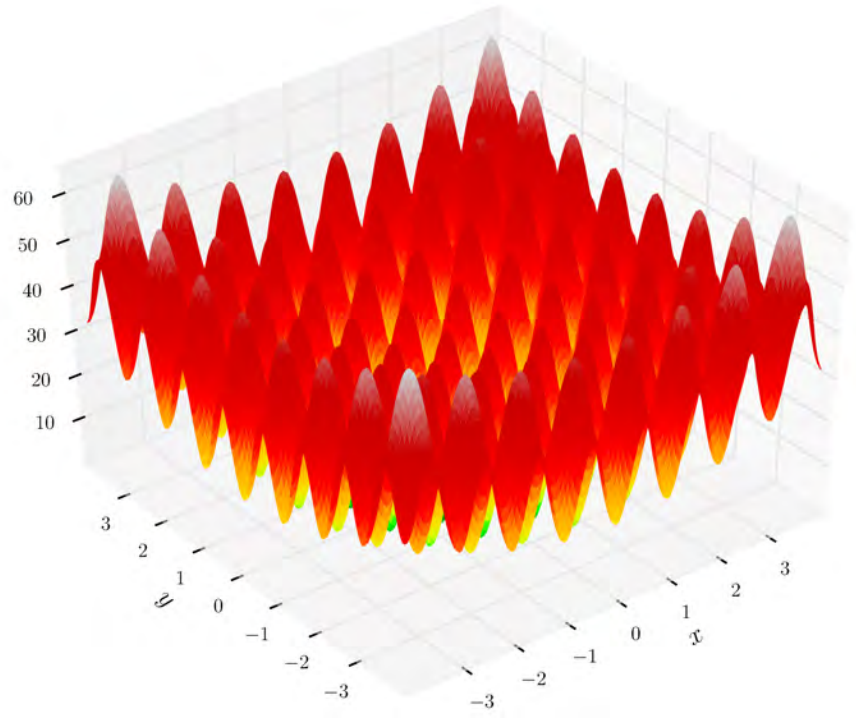
#### Himmelblau's function

Fig. **4-6**(a) shows the resultant population after the real–parameter GA was applied to optimize the Himmelblau's function, which is given by:

$$f(\boldsymbol{x}) = \left(x_1^2 + x_2 + 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2.$$




(a) Himmelblau's function



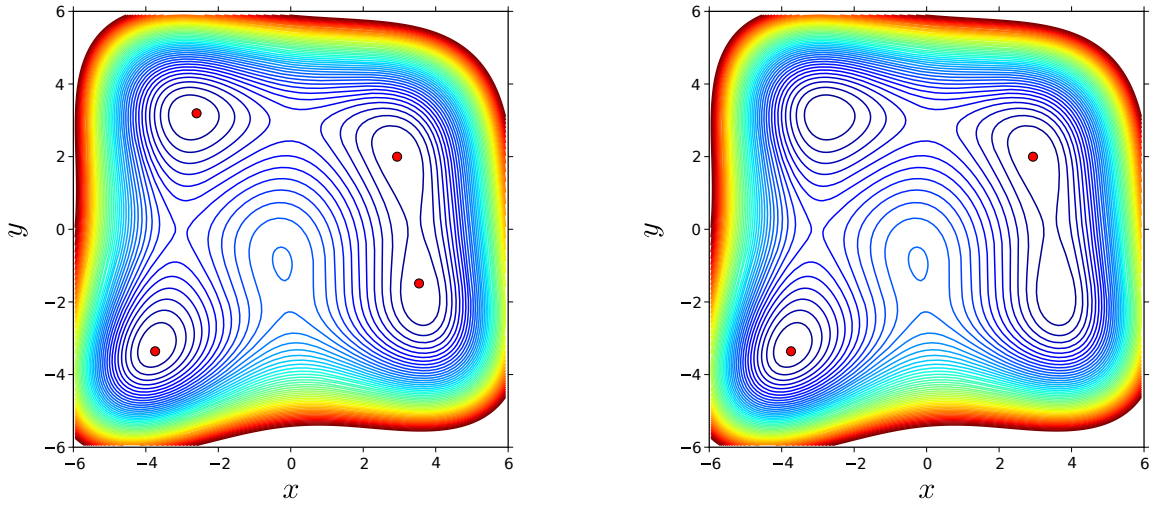

(b) Rastrigin's function

**Figure 4-5**.: Test functions

This function has 4 minima, namely:

$$\begin{aligned} f(3.0, 2.0) &= 0.0, \\ f(-2.805118, 3.131312) &= 0.0, \\ f(-3.779310, -3.283186) &= 0.0, \\ f(3.584428, -1.848126) &= 0.0. \end{aligned} \tag{4-2}$$

The red points in Fig. **4-6**(a) represent regions where individuals were concentrated at the end of the simulation, it can be seen that those points agree with those given in Eq. (4-2), this means that the algorithm is able to find the global optimum (even if there were more than one optimum - local or global). It has to be said that not all the four minima had the same quantity of individuals at the end of the simulation, some regions had more individuals than others, this means that the algorithm tries finding one unique minimum, and in order to achieve such goal, more generations are needed. Fig. **4-6**(b) shows the result obtained after 200 generations.




(a) Population after 100 generations

(b) Population after 200 generations

**Figure 4-6**.: Optimization Himmelblau's function using GAs

As said before, the algorithm converges to one unique minimum if the number of generations is large enough; with 200 generations, just two minima are identified, and one of those minima is densely populated while the other is not, so if the algorithm keeps running, sooner or later the region with more individuals around will be identified as the unique minimum in the whole space.

### Rastrigin's function

Rastrigin's function, in its more general representation is given by:

$$f(\boldsymbol{x}) = 10 \cdot q + \sum_{i=1}^{q} \left[ x_i^2 - 10 \cdot \cos(2\pi x_i) \right],$$

where $x_i \in [-5.12, 5.12]$, $1 \leq i \leq q$. This function has a global minimum at $f(\boldsymbol{x}) = 0$, and it happens when $\boldsymbol{x} = \boldsymbol{\emptyset} \in \mathbb{R}^n$.
The case treated here is for $q = 2$, and the results obtained after 100 iterations is shown in Fig. 4-7.

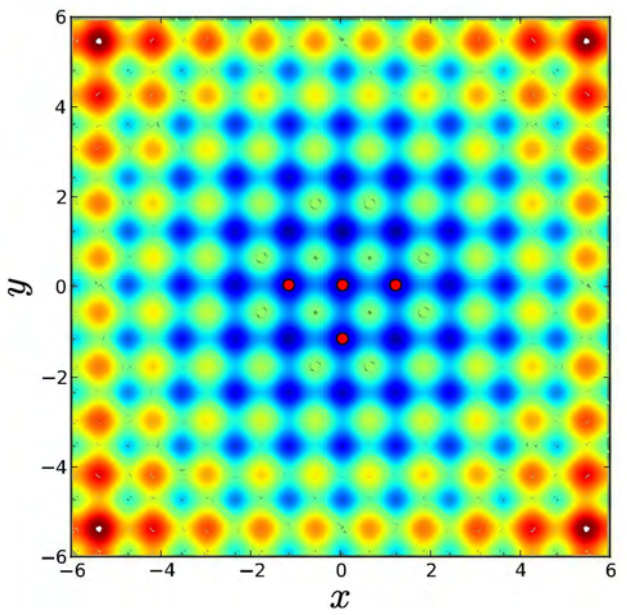



**Figure 4-7**.: Optimization Rastrigin's function using GAs – Population after 100 generations

Like in the previous examples, several minima have been identified (red points), and among them is the global optimum. An increase in the number of generations will improve the estimation in this case, because this objective function has one unique minimum (Himmelblau's function has four).

## 4.4.2. Evolution Strategies (ES)

$(\mu + \theta)$–ES were used for the simulations. The object variables $\boldsymbol{x}(t)$ were subjected to a discrete recombination, while the strategy parameters $\boldsymbol{\Sigma}, \boldsymbol{A}$ were recombined with an intermediate variation. The number of parents was fixed in 100 ($\mu = 100$), the offspring population size was also 100 ($\theta = 100$), and the algorithm performed 100 iterations (100 generations). The same functions optimized by GAs are treated by means of ES.

#### Himmelblau's function

Unlike GAs, ES converged faster towards one of the optimum points (See Fig. **4-8**). In the first simulations the algorithm had already identified the four global optima, and as time advanced, one region began to attract the individuals in other zones. The red point in Fig. **4-8** represents the optimal point identified by ES. Several runs of the algorithm shown that any point in Eq. (4-2) can be estimated, and just density of individuals in a neighborhood of optimum points will determine the convergence region towards individuals will be redirected.

#### Rastrigin's function

Again, ES were able to find the global optimum in less than 100 iterations (see Fig. **4-9**), what shows the effectiveness and good performance of this technique when optimizing single–objective problems with one global optimum and several global optima. Once several points stay in a neighborhood of the optimal point, the other members of the population will be attracted given the selection scheme used.

## 4.5. Concluding remarks about EAs

- Even if EAs are capable to find the solutions just by generating a random population in the parameter space $\mathcal{D}$ and applying evolutionary operators, a previous knowledge about the topology of the objective function is useful, because the space can be narrowed in order to accelerate the convergence of the algorithm (definition of boundaries).

- As said in Chapter 3, gradient information is irrelevant in these algorithms. EAs are flexible and can be applied to problems with one global optimum (uni–modal problem) or several global optima (multi–modal problem).

- Several single–objective optimization problems were tested using GAs and ES. Most of these artificial landscapes can be found in Appendix A. Both algorithms were able to find the global optimum in the test functions. ES shown a better performance and faster convergence rate than GAs due to the inclusion of strategy parameters that improve the search capabilities of the algorithm. Next, graphs illustrating the evolution of the object variables $(x_1, x_2)$ and the error between the objective function and the real global optimum of one of the test functions analyzed with both algorithms are shown.

  From Fig. **4-10** it can be seen that ES, in less than 15 generations, had already reached a neighborhood of the global optimum, and the error in such iteration was approximately about 0.6, which is relatively small. The error decays slowly, and at iteration 100 the magnitude of the error is approximately $7 \cdot 10^{-4}$, which is more than acceptable. On the other hand, GA had also reached a neighborhood of the global

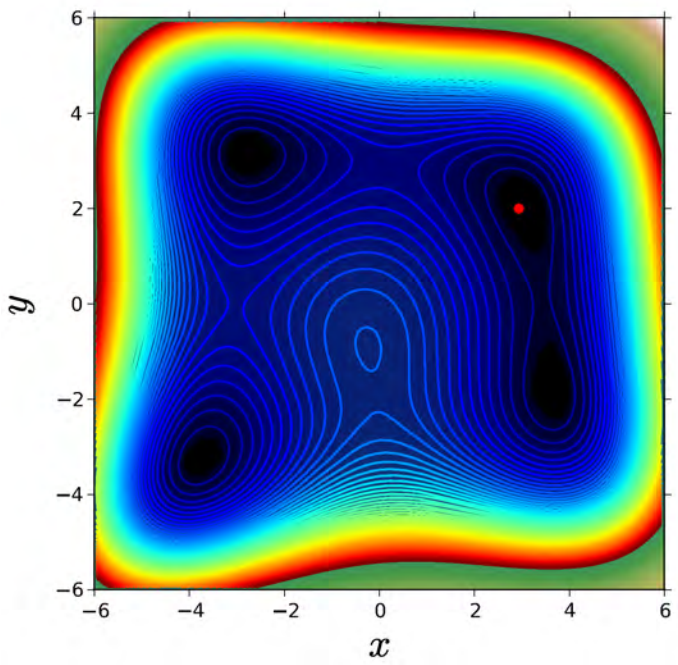



**Figure 4-8**.: Optimization Himmelblau's function using ES – Population after 100 generations

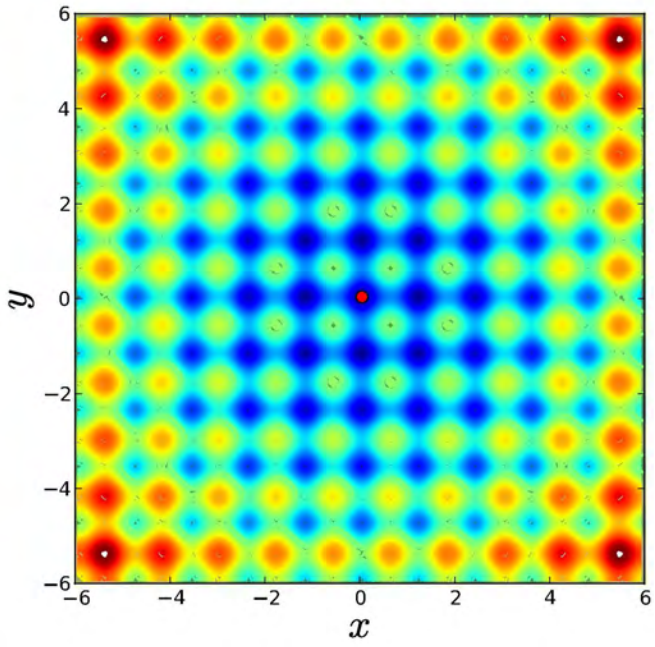
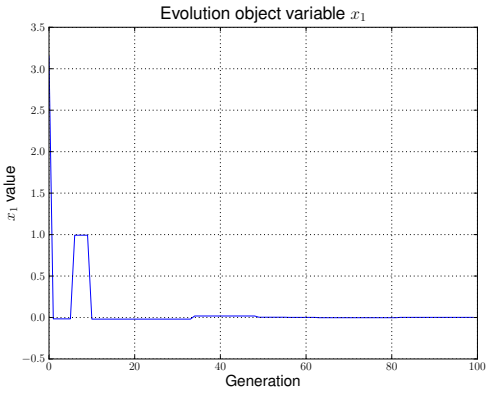
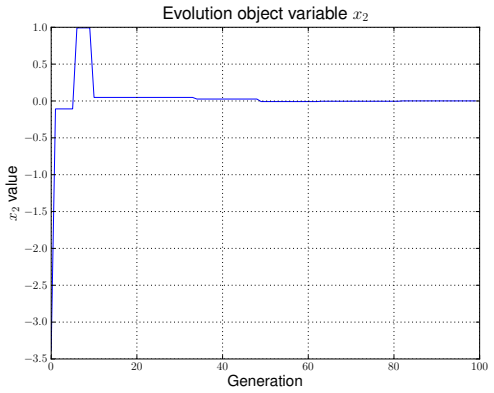


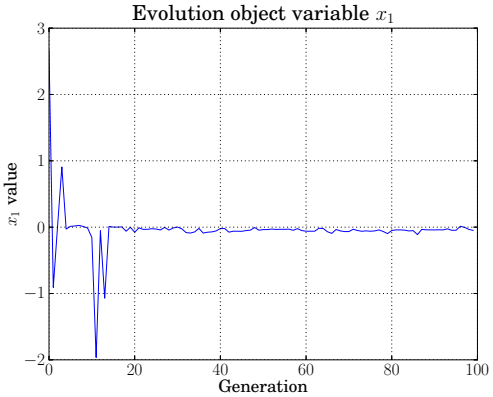

**Figure 4-9**.: Optimization Rastrigin's function using ES – Population after 100 generations



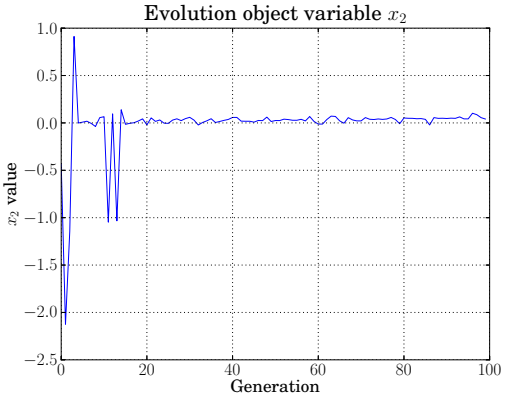

(a) Evolution object variable $x_1$ – ES




(b) Evolution object variable $x_2$ – ES

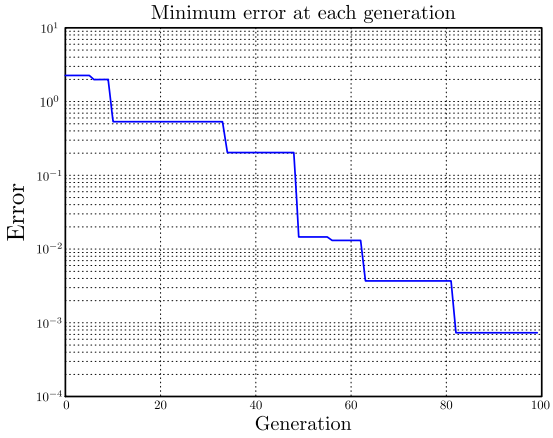



(c) Evolution object variable $x_1$ – GAs




(d) Evolution object variable $x_2$ – GAs

**Figure 4-10**.: Evolution object variables – Rastrigin's function




(a) Evoluton Strategies (ES)


(b) Genetic Algorithms (GA)

**Figure 4-11**.: Evolution error between objective function and real optimum – Rastrigin's function

optimum in less than 15 generations, but the error is greater than the one computed with ES (see Fig. **4-11**).

- These algorithms were applied to single–objective optimization problems, but the generalization to handle multi–objective optimization problems is not difficult. Next chapter addresses this issue, and it will be seen that the algorithms are basically the same, just the inclusion of some operators are needed to tackle more than one objective.

# 5. Multi–objective optimization

Next a brief description about multi–objective optimization is presented herein. The reader is referred to [33] and [31] for a complete survey on the topic.
The Chapter starts with an introduction to multi–objective optimization (MOO) problems, and introduces the concepts of *dominance* and *pareto optimality*, which are fundamental in the theory of MOO problems (Section 5.1). Section 5.2 describes the methodology used to solve this type of optimization problems. Some examples are given in Section 5.3. Finally, Section 5.4 gives some concluding remarks.

## 5.1. Basic principles of multi–objective optimization

Optimization problems involving more than one objective function are known as multi-objective optimization problems. Many practical applications fit in this type of problems, since the interaction between the parameters of the model under study must satisfy several objective functions at the same time.
In its general form, a MOO problem can be stated as Eq. (3-1), where

$$\begin{aligned} \boldsymbol{f} &: \mathbb{R}^q \to \mathbb{R}^m \\ \boldsymbol{x} &\mapsto \boldsymbol{f}(\boldsymbol{x}), \end{aligned}$$

here $\boldsymbol{f}$[1] unlike single–objective optimization problems, is a vector of $m$ components, whose elements are the objective functions to be optimized simultaneously.
Generally, there are conflicting scenarios among the objective functions involved in the optimization process, i.e., while one objective has been optimized (minimized or maximized), other objective functions might not; in consequence, when multiple conflicting objective functions arise, it is not possible to characterize a single optimum solution which optimizes all functions simultaneously; instead, a set of optimal solutions is identified. All points belonging to such a set of optimal solutions are known as *pareto optimal solutions*, and this set is called the *pareto*[2] *optimal front*. In general, there are infinite number of pareto optimal

[1] From here it will be assumed, without loss of generality, that the objective functions have the following form: $\boldsymbol{f}(\boldsymbol{p})$, so the functions will depend on a vector $\boldsymbol{p} \in \mathcal{D} \subseteq \mathbb{R}^q$.

[2] After the Italian engineer, sociologist, economist and philosofer Vilfredo Pareto (1848-1923), who made important contributions to economics, particularly in the study of income distribution and in the analysis of individuals' choice

solutions and the final goal in MOO problems is to identify the optimal pareto front. Before explaining what the pareto front is, the concept of *domination* must be introduced.

### 5.1.1. The concept of Domination

Consider the points shown in Fig. **5-1**, which represent four solutions to the multi–objective minimization problem of two objective functions $f_1(\boldsymbol{p})$ and $f_2(\boldsymbol{p})$.

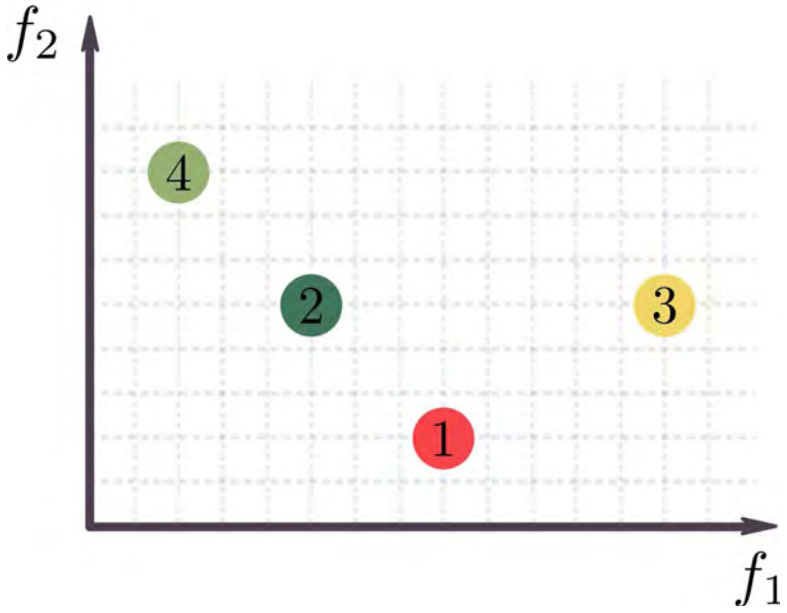



**Figure 5-1**.: Dominance in multi–objective optimization problems

Consider points 1 and 3; comparing these solutions it can be seen that solution 1 minimizes $f_1(\boldsymbol{p})$ better than solution 3, and the same conclusion can be made with respect to function $f_2(\boldsymbol{p})$. In this case it is said that solution 1 *dominates* solution 3, because such a point optimizes better both objectives than point 3. Now consider points 2 and 3; in this case the objective function $f_1(\boldsymbol{p})$ is optimized better by point 2, but both points minimize $f_2(\boldsymbol{p})$ the same way; here it is said that solutions 2 dominates solution 3, because even if both solutions minimize an objective function to the same extent, point 2 is better than point 3 with respect to the other objective.

With the previous explanation, it can be said that a solution $\boldsymbol{p}_i \in \mathcal{D}$ is said to *dominate* the solution $\boldsymbol{p}_j \in \mathcal{D}$ when $f_k(\boldsymbol{p}_i) \leq f_k(\boldsymbol{p}_j)$ for all $k = 1, \ldots, m$, and for $j = 1, \ldots, \mu$ ($\mu$ is the population size), and there exists at least one objective function $r$ which satisties $f_r(\boldsymbol{p}_i) < f_r(\boldsymbol{p}_j)$ for $j = 1, \ldots, \mu$.

Both conditions must be satisfied in order to say that solution $\boldsymbol{p}_i$ dominates solution $\boldsymbol{p}_j$; if one condition is not fulfilled, then solution $\boldsymbol{p}_i$ does not dominate solution $\boldsymbol{p}_j$. It is worth noting that if solution $\boldsymbol{p}_i$ does not dominate $\boldsymbol{p}_j$, it does not necessarily mean that solution $\boldsymbol{p}_j$ dominates $\boldsymbol{p}_i$, for instance, consider points 1 and 4 in Fig. **5-1**, it can be seen that one

point minimizes one objective function better than the other point, but with respect to the other objective the situation is completely different; in this situation it is said that points 1 and 4 are *non–dominated* to each other.

### 5.1.2. Pareto optimal front

The pareto optimal front is the set whose elements are non–dominated with respect to each other in the entire search space $\mathcal{Z}$. Fig. **5-2** shows the pareto front corresponding to a multi–objective minimization problem involving two objective functions.

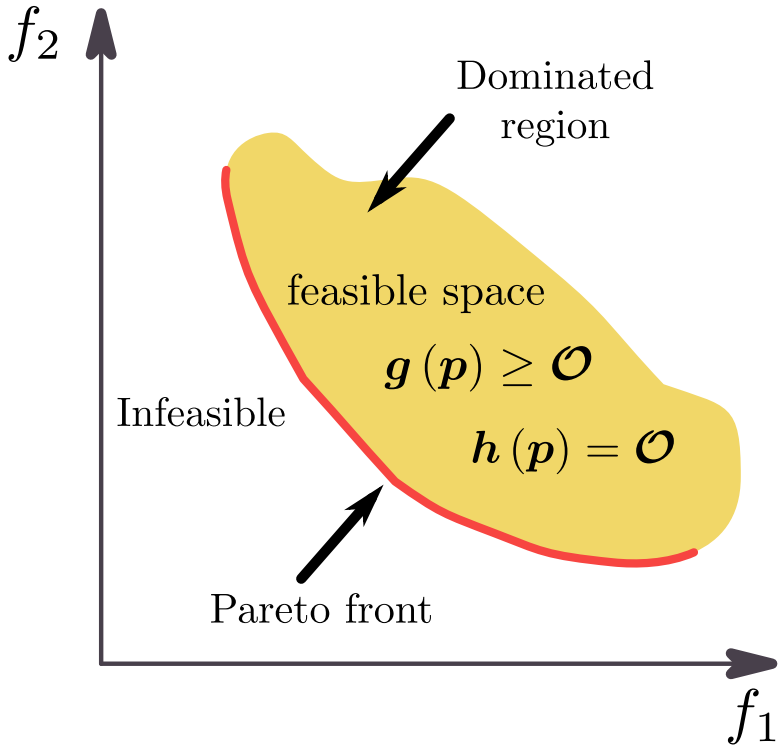



**Figure 5-2**.: Pareto front (Adapted from [70])

The feasible space represents the region where the constraints are not violated, and all the points inside the feasible space that do not belong to the pareto front are known as dominated solutions. The remainder of the search space is the infeasible space, i.e., solutions that violate the restrictions of the MOO problem.

Just as single–objective optimization, if the functions $f_k(\boldsymbol{p})$, $1 \leq k \leq m$, are multi–modal, there will be, in this case, *local pareto fronts*, which are set of points that are pareto optimal in a region of the search space, but not in the whole space. As evident, the methods used to solve MOO problems look for the global pareto optimal front, i.e., the pareto front with the best trade–off among objective functions.

## 5.2. Methods for solving multi–objective optimization problems

Several methodologies have been proposed to tackle MOO problems. There are basically two main branches:

- **Classical methods:** This methods transform MOO problems into single–objective optimization problems by means of user–defined parameters (weight vectors, $\varepsilon$ vectors, etc.). The main drawback related with these approximations is the tuning of parameters; given that the original problem has been posed as a single–objective optimization problem, in order to find several solutions from the pareto optimal front, the algorithm must be executed several times with different parameters that must be tuned by the user. Examples of this methods are:
  - Weighted sum method [71, 33].
  - $\varepsilon$–constraint method [71, 33].
  - Weighted metric method [71, 33].
  - Benson's method [15, 33].
  - among others.

- **Multi–objective evolutionary algorithms (MOBEAs):** As the name suggests, these are methods based on evolutionary algorithms (EA). Given the necessity to find a set of optimal solutions, these methods are suitable to handle these problems given its work principles (population–based methods), because just some minor modifications to the algorithms seen in Chapter 4 are needed in order to solve MOO problems. Furthermore, the user has to specify little or no information at all about the present problem, so the necessity of tuning parameters is eliminated. Some of the most important methods in this category are:
  - Vector Evaluated Genetic Algorithm (VEGA) [83].
  - Multiple Objective Genetic Algorithm (MOGA) [39].
  - Non–Dominated Sorting Genetic Algorithms (NSGA, NSGA–II) [88, 34].
  - Strength Pareto Evolutionary Algorithm (SPEA, SPEA2) [102, 103].
  - among others.

This Section will describe the *Elitist Non–Dominated Sorting Genetic Algorithm (NSGA–II)*. For more information about the other algorithms, please refer to the literature exposed at the beginning of the Chapter.

### 5.2.1. Elitist Non–Dominated Sorting Genetic Algorithm (NSGA–II)

The main idea behind this algorithm (and the other MOBEAs) is the preservation and diversification of good solutions in the whole parameter space $\mathcal{D}$. Both characteristics are important because if good solutions are found, they must be preserved during the whole process, and diversity is needed in order to spread the solutions in the parameter space and to completely identify the pareto optimal front.
In the following, we will succinctly describe the NSGA-II procedure (the reader is referred to [34] for an in-deep explanation). The original algorithm presented herein uses GAs as engine search (it could be in both fashions: binary-coded or real-parameter), but ES can be implemented instead. The steps followed by the NSGA–II procedure are outlined below:

1. Define a set of $m$ objective functions that will be gathered under the vector $\boldsymbol{f}(\boldsymbol{p}) = [f_1(\boldsymbol{p}), \ldots, f_m(\boldsymbol{p})]^T$.

2. Initialize the initial population $\boldsymbol{P}(0) = \{\boldsymbol{p}_i \in \mathcal{D} : 1 \leq i \leq \mu\}$, where $\mu$ is the number of parents, and set $t = 0$.

3. Compute for each individual $\boldsymbol{p}_i \in \boldsymbol{P}(t)$ all the objective functions, that is, calculate $\boldsymbol{f}(\boldsymbol{p}_i)$ for all $i = 1, 2, \ldots, \mu$.

4. Apply genetic operators (natural selection, crossover and mutation) in order to generate an offspring population $\boldsymbol{P}''(t) = \{\boldsymbol{p}_i'' : 1 \leq i \leq \theta\}$ from the parent population $\boldsymbol{P}(t)$; here $\theta$ stands for the offspring population size.

5. Compute for each individual $\boldsymbol{p}_i'' \in \boldsymbol{P}''(t)$ all the objective functions, that is, calculate $\boldsymbol{f}(\boldsymbol{p}_i'')$ for all $i = 1, 2, \ldots, \theta$.

6. This algorithm introduces *elitism* in its procedure, which means that the offspring population and the parent population are gathered in a single group $\boldsymbol{P}(t) \cup \boldsymbol{P}''(t)$.

7. Before selecting the next generation of parents $\boldsymbol{P}(t+1)$ the following sorting routine is performed:

   a) Identify different pareto fronts in the whole objective function space (optimal and local). Each pareto front has elements that are non-dominated to each other inside the pareto. The best pareto front of the current population will be the first front ($\mathcal{F}_1$); the next set of solutions that are just dominated by the first front will be the second front ($\mathcal{F}_2$); the third front ($\mathcal{F}_3$) has points that are dominated by the first and second front, and so on. Up to $n$ fronts ($\mathcal{F}_n$) will be identified at the end of this process. This procedure is known as *non-dominated sorting*.

   b) Now the new parent population is initialized to the empty set, that is $\boldsymbol{P}(t+1) = \emptyset$. Let $|\mathcal{F}_j|$, $1 \leq j \leq n$, be the number of elements in the pareto front $j$, and $|\boldsymbol{P}(t+1)|$ the number of elements in the parent population at time $t+1$. Set a

counter $j = 1$; while $|\boldsymbol{P}(t+1)| + |\mathcal{F}_j| \leq \mu$ make $\boldsymbol{P}(t+1) = \boldsymbol{P}(t+1) \cup \mathcal{F}_j$ and set $j = j + 1$.

c) If $|\boldsymbol{P}(t+1)| = \mu$, skip step 7.d and go to step 8.

d) Since $|\boldsymbol{P}(t+1)| + |\mathcal{F}_j| > \mu$, perform a *crowding-sort* procedure in $\mathcal{F}_j$ in order to choose only $\mu$ elements out of $\boldsymbol{P}(t+1)$. Such procedure is described next:

i. Let $r = |\mathcal{F}_j|$.

ii. For each element in the pareto front $\mathcal{F}_j$ set $d_i = 0$, $1 \leq i \leq r$. In the variable $d_i$ the so-called *crowding distance* will be stored.

iii. Sort the elements in $\mathcal{F}_j$ in ascending order according to each objective function $f_k(\boldsymbol{p})$, $1 \leq k \leq m$ and store the sorted indexes in a vector $I_k$. Here we will consider $I_k(i)$ as the $i$-th element in vector $I_k$.

iv. For each objective function, $f_k$, $k = 1, \ldots, m$, make:

$$d_{I_k(1)} = \infty$$

$$d_{I_k(i)} = d_{I_k(i)} + \frac{f_k\left(\boldsymbol{p}_{I_k(i+1)}\right) - f_k\left(\boldsymbol{p}_{I_k(i-1)}\right)}{f_k\left(\boldsymbol{p}_{I_k(r)}\right) - f_k\left(\boldsymbol{p}_{I_k(1)}\right)} \quad \text{for } i = 2, \ldots, r-1$$

$$d_{I_k(r)} = \infty$$

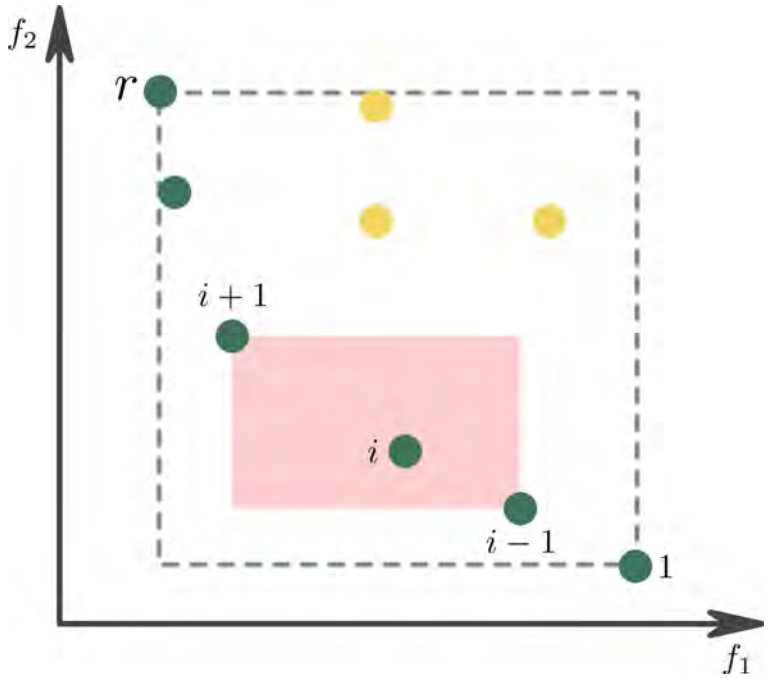



**Figure 5-3**.: Crowding distance calculation (Adapted from [33])

Consider the minimization of two objective functions $f_1(\boldsymbol{p}), f_2(\boldsymbol{p})$ shown in Fig. **5-3**. The green points represent the pareto front $\mathcal{F}_j$. It can be seen that

the crowding distance of the point $i$ is the semiperimeter of the hypercube whose vertices are the neighboring solutions $i-1$ and $i+1$ (shaded area in Fig. **5-3**), normalized by the semiperimeter of the hypercube enclosed by the boundary solutions (1 and $r$).

From the resulting sorted pareto front, select the remaining $\mu - |\boldsymbol{P}(t+1)|$ elements with the largest *crowding distances* $d_i$. The selection of the largest crowding distance is intended to preserve diversity among solutions, because a large crowding distance means a point whose neighbors are far from it, and consequently, the point is located in a less crowded region.

8. Set $t = t + 1$

9. Repeat steps 4 to 8 until the pareto optimal front has been fully identified or a termination criterion is met.

## 5.3. Examples

Two examples taken from [33] are presented herein in order to show the capabilities of the NSGA–II procedure when dealing with multiple objectives. Some test functions for multi–objective optimization are given in Appendix A.

### 5.3.1. Example minimization problem

Consider the following bi-objective minimization problem:

$$\text{Minimize: } \begin{cases} f_1(x,y) & = x \\ f_2(x,y) & = \frac{1+y}{x} \end{cases} \tag{5-1}$$

for $0.1 \leq x \leq 1$, $0 \leq y \leq 5$.

In this case, there are not equality constraints $\boldsymbol{h}(\boldsymbol{x})$, and the boundaries of the values can be used as inequality constraints $\boldsymbol{g}(\boldsymbol{x})$.
To solve this problem, the NSGA–II procedure with real–parameter GAs as EA is employed. The following parameters were set up:

- Parent population size $\mu = 200$.
- Offspring population size $\theta = 200$.
- Number of generations: 500.

- Simulated binary crossover, with crossover distribution index $\eta_c = 15$, was used as mechanism to generate new members; and for mutation, the polynomial variant with mutation distribution index $\eta_m = 20$ was applied.

From Fig. **5-4**(b) it can be seen how the NSGA–II procedure is able to find the optimal pareto front and keep the diversity among the solutions. The last characteristic is due to the crowding distance used in the selection stage of the algorithm. The mechanism of elitism is also important in the implementation of the algorithm because it prevents the removal of good solutions from the parent population that have been found in the first generations.

### 5.3.2. Practical application: Design of a cantilever beam

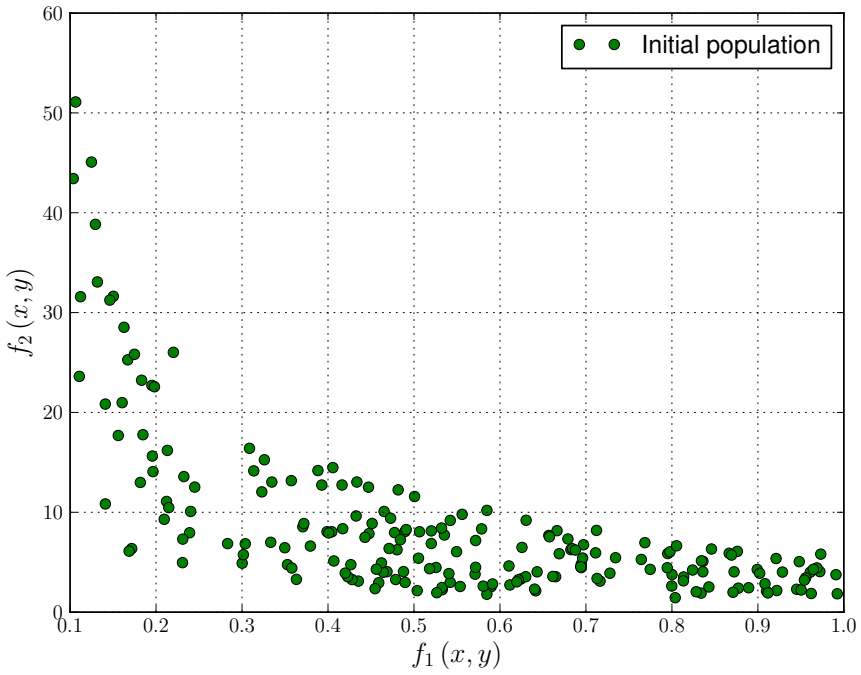
In this case a typical problem from structural design is solved by means of MOBEAs. Consider the cantilever beam shown in Fig. **5-5**.
The beam has to support a load $P$ applied at the end of the beam. The idea is try minimizing the weight ($f_1(\phi, L)$) and the deflection ($f_2(\phi, L)$) of the beam, which are stated as function of the diameter ($\phi$) and the length ($L$) of the beam. It can be seen that both objectives are conflicting each other, because in order to avoid large deflections, the dimensions of the beam necessarily must be large, but with an increase of $\phi$ and $L$, the weight goes up too, which is contradictory according to the objective of minimizing the weight of the beam.
The optimization problem is stated as a bi-objective minimization problem with two constraints, which are the maximum allowable stress in the beam ($\sigma_{\max}$) and the maximum permissible deflection ($\delta_{\max}$):

$$\text{Minimize: } \begin{cases} f_1(\phi, L) & = \rho \frac{\pi \phi^2}{4} L \\ f_2(\phi, L) & = \frac{64 P L^3}{3 E \pi \phi^4} \end{cases} \tag{5-2}$$

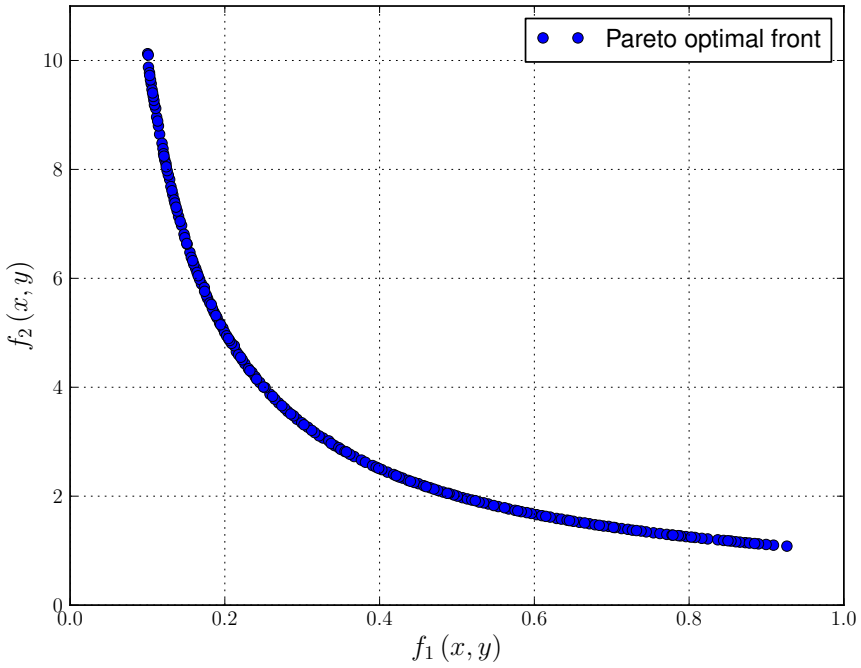
$$\text{subject to } \begin{cases} \sigma_y & \leq \sigma_{\max} \\ \delta & \leq \delta_{\max} \end{cases}$$

for 10 mm $\leq \phi \leq$ 50 mm, 0.20 m $\leq L \leq$ 1.00 m. In Eq. (5-2) the following parameters are used:

- Density of the material ($\rho$): 7800 kg/m$^3$.
- Applied load ($P$): 1 kN.
- Elastic modulus of the material ($E$): 207 GPa.
- Stress exerted on the cantilever beam ($\sigma_y$): The maximum stress is calculated as follows:
  $\sigma_y = \dfrac{32PL}{\pi \phi^3}$.




(a) Initial population




(b) Pareto optimal front after 500 generations

**Figure 5-4**.: Example bi–objective minimization problem (Eq. (5-1))

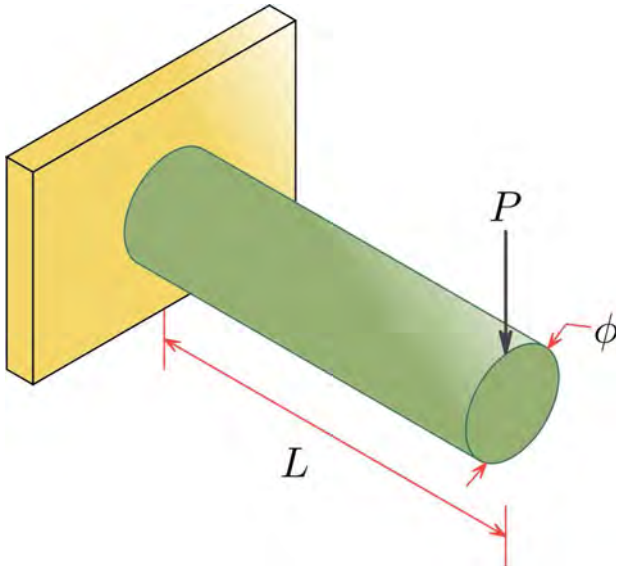


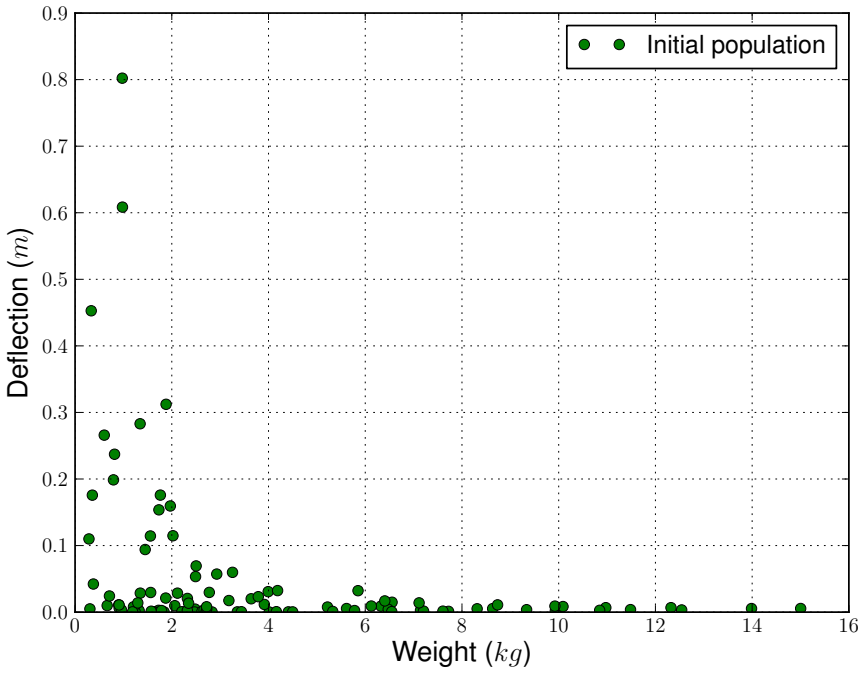

**Figure 5-5**.: Cantilever beam subjected to load $P$

- Maximum allowable stress ($\sigma_{\max}$): 300 MPa.
- Maximum permissible deflection ($\delta_{\max}$): 5 mm.

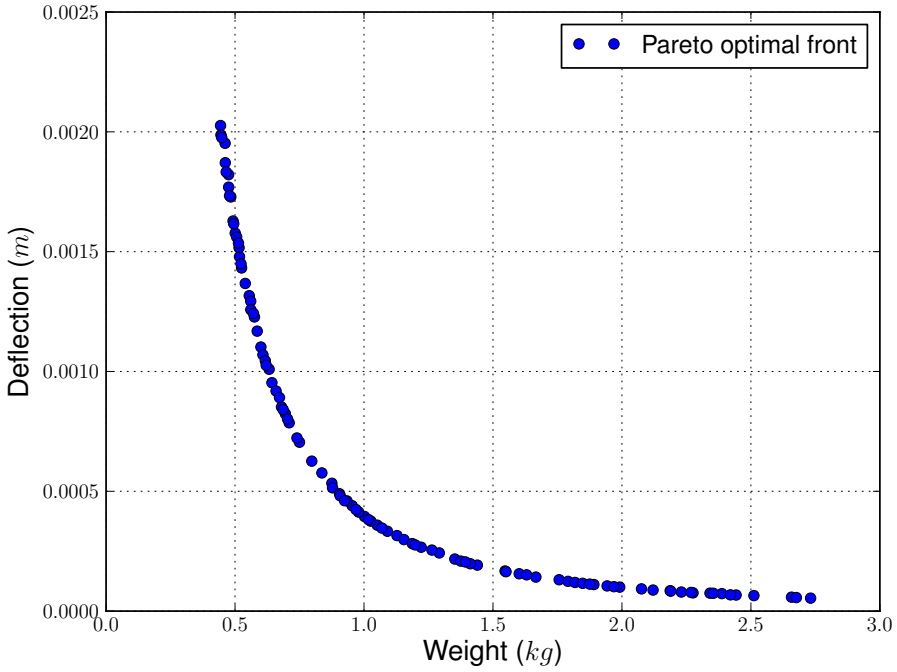
Fig. **5-6**(a) shows the initial population used by NSGA–II. It can be seen that most of the individuals violate the second objective (deflection) which is supposed to be less or equal than 5 mm. The parameters of the NSGA–II procedure were set up with the following values:

- Parent population size ($\mu$): 100.
- Offspring population size ($\theta$): 100.
- Number of generations: 200.
- Simulated binary crossover, with crossover distribution index $\eta_c = 15$.
- Mutation via polynomial variant with mutation distribution index $\eta_m = 20$.

After the algorithm is executed, the pareto shown in Fig. **5-6**(b) is the final result. As in the previous example, diversity of solutions is preserved throughout generations, so a set of optimal solutions (pareto optimal front) is obtained at the end of the simulation. The algorithm can identify the pareto optimal solutions and preserve in that region all the generated solutions.
Now is task of the designer to choose one solution among the final population of non–dominated solutions; this task is achieved only if higher–level information is available, because without complementary information is difficult to said that one solution is better than other if both are part of the pareto optimal front.




(a) Initial population




(b) Pareto optimal front after 200 generations

**Figure 5-6**.: Design of the cantilever beam analyzed in Section 5.3.2

## 5.4. Summary and concluding remarks

- Optimization problems with two or more objectives are known as multi–objective optimization (MOO) problems. Given the conflicting nature between objective functions, a unique solution is not readily identifiable, so a set of optimal solutions (Pareto optimal front) must be devised.

- There are two mainstreams for solving MOO problems, namely:
  - **Classical methods:** These methods try solving the problem by finding a single optimal solution, which is accomplished by means of a transformation of the MOO problem into a single–objective optimization problem. The main drawback related with these methods is the necessity of the user to adjust parameters in order to find solutions in the pareto optimal front, and those parameters must be tuned every time a new solution is required.
  - **Multi–objective evolutionary algorithms (MOBEAs):** These algorithms work with a population of solutions instead of single points, so several solutions can be computed in a single run. EAs are used as search engine in the parameter space, so derivatives are not computed; furthermore, the user neither needs to specify information related with the topology of the problem nor the parameters that must be tuned.

- MOBEA procedures must satisfy two conditions:
  - Preserve good solutions throughout the process in order to increase the probability of reaching the pareto optimal front in a few number of iterations.
  - Maintain the diversity of solutions among the population so the pareto optimal front will be easily identified.

- The NSGA–II procedure is suitable to solve MOO problems given its elitist approach and its low computational complexity. The algorithm is easy to program and it is able to handle problems that have continuous or discontinuous pareto optimal fronts. Furthermore, the NSGA–II technique satisfies the two conditions listed above due to the implementation of the *elite–preserving* operator and the *crowded tournament* selection operator.

- After a pareto optimal front has been identified, the designer has to select a unique solution from such a set, because all solutions cannot be used in practical applications. There are methods used to reduce the number of solutions in the pareto optimal front to a few number of solutions, so a better representation of the pareto frontier is achieved. For information related with those methodologies, the reader is referred to [33]. It is worth noting that, most of the time, the designer has an idea about the problem at hand, so even if these methods are applied, the final solution will reflect the designer's

preference, so a previous knowledge about the system or model under study provides tools that ease the decision making process.

- Several examples are given in Appendix A, where the objective functions to optimize are stated, and the pareto optimal fronts of each example are plotted. It can be seen how to NSGA–II algorithm can find the pareto frontier with a good distribution of solutions along the pareto optimal front.

- Next Chapter will give a brief description about several methodologies used for the identification of hysteretic structural systems. Then, a novel methodology that treats the problem of identification of hysteretic structural systems by means of MOBEAs is presented, and the algorithm to use will be the NSGA–II procedure.

# Part III.

# Identification of hysteretic structural systems

# 6. Literature review

Several methods have been proposed for the identification of the parameters of the BW-type models presented in Chapter 2, using experimental input and output data. The procedures suggested in the literature to tune the parameters of the BW-type models can be classified into two groups as (see [55]):

- Methods based on the minimization of a loss function, using for example least square estimation (see e.g. [41]), Gauss-Newton methods like the Levenberg-Marquardt algorithm (see e.g. [101]), evolutionary algorithms (see e.g. [24, 43]), differential evolution (see e.g. [64, 95]), particle swarm optimization (see e.g. [100, 23]), the generalized reduced gradient method (see e.g. [101]), etc.; in this case, the error difference between the time histories (displacements) is minimized.

- Procedures based on nonlinear filtering, using for example the extended Kalman filter (see e.g. [101, 99]), the unscented Kalman filter (see e.g. [96, 25, 26, 98]), or particle filters (see e.g. [65, 66, 25]).

This Chapter intends to give a brief description of some of the methods listed above, showing their advantages and disadvantages when computing the parameters of BW-type models of hysteresis.

## 6.1. Methods based on the minimization of a loss function

### 6.1.1. Least square estimation

In this method it is assumed that the mass of the system ($m$) and the ground acceleration ($u(t)$) are known. Eq. (2-8) can be rewritten as:

$$(\ddot{x}(t) - u(t)) = \begin{bmatrix} -x(t) & -\dot{x}(t) \end{bmatrix} \begin{bmatrix} \alpha\omega_0^2 \\ 2\xi\omega_0 \end{bmatrix} + \left(-(1-\alpha)\,\omega_0^2 z(t)\right), \tag{6-1}$$

here $x(t), \dot{x}(t)$ and $\ddot{x}(t)$ stand for the displacement, velocity and acceleration of the structure respectively, $z(t)$ is the hysteretic displacement, $u(t)$ is the external excitation exerted on the system, $\omega_0$ is the pseudo-natural frequency of the structure, $\xi$ is the viscous damping ratio

and $\alpha$ is the ratio of post-yield $k_f$ to pre-yield (elastic) $k_i$ stiffness (see Eq. (2-7)). Eq. (6-1) can be stated as:

$$y = \boldsymbol{x}^T \boldsymbol{\phi} + d,$$

where:

$$\begin{aligned} y &= \ddot{x}(t) - u(t), \\ \boldsymbol{x} &= \begin{bmatrix} -x(t) \\ -\dot{x}(t) \end{bmatrix}, \\ \boldsymbol{\phi} &= \begin{bmatrix} \alpha\omega_0^2 \\ 2\xi\omega_0 \end{bmatrix}, \\ d &= -(1-\alpha)\,\omega_0^2 z(t). \end{aligned}$$

The estimation $\hat{y}$ of the theoretical output $y$ is computed as:

$$\hat{y} = \boldsymbol{x}^T \hat{\boldsymbol{\phi}} + \hat{d},$$

where $\hat{\boldsymbol{\phi}}$ and $\hat{d}$ are the estimates of $\boldsymbol{\phi}$ and $d$ respectively. These estimates are computed according to the following equations:

$$\begin{aligned} e &= y - \hat{y}, \\ \dot{\hat{\boldsymbol{\phi}}} &= -g\boldsymbol{\Sigma}\boldsymbol{x}e - \kappa\boldsymbol{\Sigma}\,|e|\,\hat{\boldsymbol{\phi}}, \\ \dot{\hat{d}} &= -ad - g_1 e, \\ \dot{\boldsymbol{\Sigma}} &= -\boldsymbol{\Sigma}\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{\Sigma}, \end{aligned} \tag{6-2}$$

where $e$ is the estimation error, $\boldsymbol{\Sigma}$ is the covariance matrix, and $a$, $\kappa$, $g$ and $g_1 = (2y - 1)$ are positive constants that need to be tuned.

Eq. (6-2) are the recursive least-squares algorithm, which has been used for the identification of structural parameters (such as damping and stiffness) of structures subjected to random vibrations, and also to compute the hysteretic response of the system using the BW differential hysteresis model [41].

The method has some advantages, such as easy implementation in computer, the algorithm is *stable* in the sense that the hysteretic component error $\hat{d}$ is bounded, and small errors are obtained provided good initial estimates of the constants and covariance matrix listed above. On the other hand, there are some disadvantages related with this algorithm, mainly the necessity to define an initial covariance matrix and a set of parameters ($a$, $g_1$, $\kappa$), that must be tuned in order to compute satisfactorily the response of the system.

### 6.1.2. Gauss-Newton methods

The Gauss-Newton method is used to minimize a sum of squared function values that do not require second derivatives, which can be challenging to compute. This method tries to minimize the integrated mean square error ($Q$), defined as:

$$Q = \frac{1}{T}\int_0^T e^2 \mathrm{d}t,$$

where $e = \hat{\ddot{x}}(t) - \ddot{x}(t)$ for the case of structural identification. Here $T$ stands for the sampling time. The minimization is performed with respect to the parameters of the analyzed BW-type model of hysteresis.
Basically, the Gauss-Newton iterative procedure uses the following update rule:

$$\boldsymbol{\phi}_{p+1} = \boldsymbol{\phi}_p - \boldsymbol{H}_p^{-1}\nabla Q_p,$$

where $\boldsymbol{\phi}$ is the set of parameters to be identified, $\boldsymbol{H}_p$ is the Hessian matrix of $Q$, $\nabla Q_p$ is the Jacobian vector of $Q$ and $p$ is the iteration index.
For the case of identification of BW-type models, it is advisable to divide the identification process in several stages, i.e., at the beginning some parameters are kept fixed, while the other parameters are allowed to vary while minimizing the error $Q$; after the minimization of Q is performed, some parameters that were held fixed in the previous simulation are now allowed to vary with the other parameters, other parameters still remain fixed. This proces is repeated until all parameters are used for the minimization of $Q$.
This algorithm has been used for the identification of hysteretic structural systems that exhibit degradation and pinching [62], [101]. The results drawn from those researches show that the algorithm is able to estimate the parameters of the hysteresis model with a good accuracy, and it is computationally easy to implement.
Disadvantages of this method are related to the computational burden required to compute the derivatives (Jacobian and Hessian), it is required the definition of an initial state (initial guess of the parameters), and the algorithm is not well-suited for large systems where other techniques, such as the conjugate gradient method, may be more efficient.

### 6.1.3. Evolutionary algorithms

These algorithms are widely used in global optimization, given its ability to solve complex problems with a few or no information at all about the problem being analyzed. Evolutionary algorithms were treated in Chapter 4, where a description about the operators involved in these methodologies were explained.
In the identification of hysteretic structural systems, EAs have been used successfully and have proven to be a promising tool in the identification of this type of systems.
Charalampakis and Koumousis [24] used a *hybrid* EA composed of a *saw-tooth* GA [60] and *greedy ascent hill climbing* to local optimality [36]. Those techniques are improvements to

the standard GA introduced in Chapter 4. The objective function used in such study was the normalized mean square error of the predicted time history $\hat{x}(t|\boldsymbol{p})$,:

$$OF(\boldsymbol{p}) = \frac{\sum_{i=1}^{l} \left(x(t_i) - \hat{x}(t_i|\boldsymbol{p})\right)^2}{l\sigma_y^2}, \tag{6-3}$$

where $\boldsymbol{p}$ is the vector of parameters to be estimated, $l$ is the number of points in the time history $x(t)$ and $\sigma_y^2$ is the variance of the reference time history $x(t)$. The reference history $x(t)$ refers to the displacements or accelerations suffered by the structure being analyzed.
Other study carried out for the identification of hysteretic structural systems was performed by Kwok *et al.* [63], who removes the selection stage of GAs and proposed a termination criterion based on statistical tests that guarantees a good approximation to the optimal solution.
Another EA that has been used recently in the identification of BW-type models of hysteresis is the *differential evolution* (DE) algorithm [1], [95]. This algorithm tries to minimize a cost function, such as the one given in Eq. (6-3), through repeated cycles of evolutionary operators (selection, crossover and mutation). The difference between DE and the other EAs is the way the evolutionary operators are applied. Basically, DE transforms a randomly generated initial population of parameter vectors into an optimal solution through tournaments performed between each vector and *trial* vectors that are generated by mutation and crossover.
The previous studies showed that EAs are suitable for analyzing and identifying hysteretic structural systems, and the methods proved to be efficient, robust and insensitive to noise-corrupted data.
A minor disadvantage of these methods is the need to set algorithm hyper-parameters for the optimization, but recent investigations (for example [95]) have overcome this problem by adding operators that allow the algorithms to learn and adapt their own hyper-parameters throughout the optimization process.

### 6.1.4. Particle swarm optimization

Particle swarm optimization (PSO) is a population-based optimization technique that mimics the behaviour of swarms of insects and satisfies the five axioms of swarm intelligence, namely proximity, quality, diverse response, stability and adaptability [59]. This algorithm generates a random initial population of *particles* that are attracted towards better positions in the decision space $\mathcal{D}$, according to the local best known position.
Basically the method tries to move the swarm toward the best solution according to a given measure of quality (such as Eq. (6-3)). PSO moves the particles around the decision space according to simple mathematical expressions related to the *position* and *velocity* of each particle. The position of each particle refers to the model parameters to be estimated.
Charalampakis and Dimou [23] used PSO to compute the parameters of the simple BW model of hysteresis of a bolted-welded steel connection. The results of this work showed

that PSO is able to estimate the parameters of the model, and the method proved to be accurate and robust with respect to other methodologies. It must be pointed out that the methodology has not been used for the identification of BW-type models that involve the effects of degradation and pinching, that are very common in structures when subjected to seismic loads.

## 6.2. Methods based on nonlinear filtering

### 6.2.1. Extended Kalman filter

The extended Kalman filter (EKF) is an approximation of a nonlinear system into a linear system by means of a truncated first order Taylor series expansion. It applies the equations of the basic Kalman filter for linear systems, linearizing the dynamic of the model around the current state, whose distribution is assumed to be a Gaussian random variable. For a complete treatment about the theory of Kalman filters the reader is referred to [48] and [45]. The EKF has two major steps:

1. **Prediction stage:** where the actual state of the model $\boldsymbol{x}(t)$ and the covariance matrix $\boldsymbol{\Sigma}$ at time $t$ are projected at time $t+1$.

2. **Filtering stage:** the projected values of $\boldsymbol{x}(t)$ and $\boldsymbol{\Sigma}$ are adjusted according to the real response of the system at time $t+1$.

The EKF considers a nonlinear dynamical system described by the following state-space model:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}\left(k, \boldsymbol{x}_k\right) + \boldsymbol{w}_k, \\ \boldsymbol{y}_k &= \boldsymbol{h}\left(k, \boldsymbol{x}_k\right) + \boldsymbol{v}_k, \end{aligned}$$

where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent zero-mean white Gaussian noise processes, $\boldsymbol{f}\left(k, \boldsymbol{x}_k\right)$ is the nonlinear transition matrix that is possibly time-variant, and $\boldsymbol{h}\left(k, \boldsymbol{x}_k\right)$ is the nonlinear measurement matrix that may be time-variant too.
The basic idea of the EKF is to linearize the state of the model at each time instant around the current state estimate. Once a linear model is obtained, the equations of the standard Kalman filter are applied. The approximation is performed by the following two matrices:

$$\begin{aligned} \boldsymbol{F}_{k+1,k} &= \left.\frac{\partial \boldsymbol{f}\left(k, \boldsymbol{x}_k\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k}, \\ \boldsymbol{H}_k &= \left.\frac{\partial \boldsymbol{h}\left(k, \boldsymbol{x}_k\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-}, \end{aligned}$$

where $\boldsymbol{F}_{k+1,k}$ is a matrix that stores the partial derivatives of the nonlinear transition matrix $\boldsymbol{f}(k,\boldsymbol{x})$ with respect to the states $\boldsymbol{x}$, $\boldsymbol{H}_k$ is matrix that stores the partial derivatives of the nonlinear measurement matrix $\boldsymbol{h}(k,\boldsymbol{x}_k)$ with respect to the states and $k$ denotes the iteration counter. In the former case, the derivatives are evaluated at $\hat{\boldsymbol{x}}_k$ (a posteriori estimate of the state at time $k$). In the latter case, the derivatives are evaluated at $\hat{\boldsymbol{x}}_k^-$ (a priori estimate of the state at time $k$). All the entries of the matrices $\boldsymbol{F}_{k+1,k}$ and $\boldsymbol{H}_k$ are all known, by having $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{x}}_k^-$ available at time $k$.
For the case of identification of BW-type models of hysteresis involving degradation and pinching (BWBN model of hysteresis), the nonlinear transition matrix $\boldsymbol{f}(k,\boldsymbol{x})$ is given by:

$$\boldsymbol{f}(k,\boldsymbol{x}) = \begin{bmatrix} \dot{x}(k) \\ u(k) - 2\xi\omega_0\dot{x}(k) - \alpha\omega_0^2 x(k) - (1-\alpha)\,\omega_0^2 z(k) \\ h(k)\frac{A(k)\dot{x}(k) - \nu(k)\left(\beta|\dot{x}(k)||z(k)|^{n-1}z(k) + \gamma\dot{x}(k)|z(k)|^n\right)}{\eta(k)} \\ (1-\alpha)\omega_0^2 z(k)\dot{x}(k) \\ \boldsymbol{\emptyset}_{18\times 1} \end{bmatrix},$$

and the nonlinear measurement matrix $\boldsymbol{h}(k,\boldsymbol{x}_k)$ can vary depending on the quantity measured (displacements or accelerations); for the case of displacements, $\boldsymbol{h}(k,\boldsymbol{x}_k) = x(k)$, and for the case of accelerations $\boldsymbol{h}(k,\boldsymbol{x}_k) = u(k) - 2\xi\omega_0\dot{x}(k) - \alpha\omega_0^2 x(k) - (1-\alpha)\,\omega_0^2 z(k)$ [96]. The parameters used in these equations are the same as those listed in Section 2.2.2.
Modifications to the EKF algorithm have been proposed and used for the identification of hysteretic structural systems with successful results. For example, Yang and Ma [99] proposed a constrained EKF with a global weighted iteration strategy which was effective in estimating all the parameters of the BW-model of hysteresis. Zhang *et al.* [101] also applied the EKF for the identification of hysteretic systems that exhibit degradation and pinching; all the parameters of the BW-model were identified without problem.
Nonetheless, even if this technique has been applied successfully in the field of structural identification, the EKF presents numerical problems when coping with complex systems. Furthermore, the EKF linearizes the nonlinear transition matrix and the nonlinear measurement matrix, which requires the computation of Jacobians that can be difficult to compute, may introduce errors in the estimates which may lead to suboptimal performance and divergence of the filter, and increase the computational burden of the algorithm.

### 6.2.2. Unscented Kalman filter

The unscented Kalman filter (UKF) appears as an alternative to solve the problems that may arise when using the EKF. This algorithm was proposed by Julier and Uhlmann [57], and furhter developed and enhanced by Wan and Van Der Merwe [89, 90].
The difference between the EKF and the UKF lies in the way that the Gaussian random variables are represented and propagated through the system dynamics. Remember that

the EKF propagates the current state of the system through the first-order linearization of the original nonlinear system. On the other hand, the UKF does not compute Jacobians, instead it uses a set of carefully chosen points (sigma points) that are propagated through the *true* nonlinear system.

The UKF does not approximate nonlinear functions of the system and measurement equations. Instead, it approximates the posterior probability density by a Gaussian density, which is represented by the sigma points. When those points are propagated through a non-linear transformation, they capture the true mean and covariance up to the second order of any non-linearity (the EKF achieves just first-order accuracy). For a complete review of the UKF, the reader is referred to [57] and [89, 90].

In the field of structural dynamics, the UKF has been used for the identification of non-degrading and degrading hysteretic systems [96, 25, 26, 98]. The identification results show that the UKF is well suited for the identification of complicated non-linear systems and that this methodology can yield accurate estimates of the parameters of the BW-type models of hysteresis. The results also show that the UKF outperforms the EKF with regard to computational efficiency and robustness to measurement noise levels. However, a suitable initial state estimate and filter parameters must be provided in order to achieve a good solution, otherwise, suboptimal solutions will be estimated, or even, the algorithm may diverge.

As a personal note, it is worth pointing out that the author of this Thesis (Gilberto Ortiz) and the Adviser (Diego A. Álvarez) implemented the method proposed by Wu and Smyth [96] and Chatzi and Smyth [25] for the identification of the parameters of BW-type models of hysteresis independently, but both algorithms were not able to estimate the parameters, but the filters converge with other type of systems found in the literature, so the algorithms are well implemented. We tried to make contact with Prof. Smyth, but he replied that he had not time to help us, and he did not give us the source code of the algorithms presented in those papers, so we could not reproduce those researches. Other paper that uses an iterated Unscented Kalman filter was found [98]; we tried to reach the author, but he did not reply.

### 6.2.3. Particle filters

A particle filter (PF), also known as sequential Monte Carlo method, is an iterating technique based on Bayesian state estimation and Monte Carlo methods, which is able to handle any functional non-linearity and measurement noise of any distribution.

The concept of the method is that the approximation of the posterior probability state is done through the generation of a large number of samples using Monte Carlo methods (weighted particles). The particles are concentrated in regions of high probability instead of uniformly distributed over the state. As the number of samples increases, this Monte Carlo approach becomes an equivalent representation of the probability density function, and the solution approaches the optimal Bayesian estimate. For further details about this method,

the reader is referred to [3], [65, 66], [25] and [35].
PFs have been used for the analysis of the hysteretic response of MDOF systems [25], and also for the identification of BW-models that take into account the stick-slip phenomenon [66]. The results showed that this algorithm is able to compute the parameters of the model, but at the same time it was observed that the filter performs poorly in identifying the time invariant model parameters if the initial interval from which the particles were sampled did not contain the true value of the corresponding parameters [25].
The main drawback related with the PFs is the fact that depending on the problem, a large number of samples may be required, thus making the analysis computationally expensive. Other disadvantage of the method is that in order to start the identification algorithm, good initial estimates of parameters are needed, otherwise, erroneous results can be obtained.

## 6.3. Concluding remarks

- BW-type models of hysteresis are highly nonlinear and any gradient-based technique (least-squares methods, Gauss-Newton methods, etc.) tend to be trapped in local minima, and therefore may fail to converge.
- Techniques based on EAs are suitable to handle the identification of parameters of BW-type models. Up to the author's knowledge, just single objective optimization procedures have been used for the identification of this type of systems using EAs, and most of the researches address the identification of non-degrading BW-type models, or assumptions are made in order to reduce the total number of parameters to be estimated [1].
- Most of the techniques listed above need to tune parameters a priori in order to start the identification stage (for example the covariance matrix), and the convergence properties of the algorithms depend on those parameters. A good estimate of those parameters is needed in order to achieve desirable results, otherwise, poor performance of the techniques and even divergence of those will be the result.
- It is found in the literature that when one applies the UKF to a highly complex system with weak observability, the standard UKF shows its weakness in robustness, convergence speed and tracking accuracy [98]. The scenario is worst when the EKF is used in such cases. Observability is a measure used in control theory in order to check if the internal states of a system can be inferred by knowledge of the outputs of the system.
- Given the personal experience that we had with the UKF, it can be said that such methodology may not work for the identification of parameters of BW-type models of hysteresis that involve the effects of degradation and pinching, so a review of the researches performed on this topic should be done.

- PFs are a good choice when working with highly nonlinear models. The problem with this method is the computational burden required to achieve an acceptable accuracy with respect to the response of the system.

- Other researches that use other methodologies that the ones listed above have been performed in recent months. For example, Worden and Hensmann [94] adopted a Bayesian approach in order to identify the parameters of non-degrading BW-type models of hysteresis; Xie *et al.* [97] used artificial neural networks as identification method; again, the non-degrading BW-type model of hysteresis was adopted. Up to the author's knowledge, there are no recent investigations addressing the identification of degrading BW-type models of hysteresis.

- All the proposed methods for the identification of the parameters of BW-type models exposed in this chapter try to reduce only the mean square error between a reference time history $x(t)$ measured in the real structure (displacements or accelerations) and the estimated one. Next Chapter proposes a novel methodology that uses multi-objective optimization techniques, so more than one objective function is proposed in order to estimate the parameters of BW-type models of hysteresis.

# 7. Identification of hysteretic structural systems using MOBEAs

In this Chapter, MOBEAs will be used as system identification technique to estimate the unknown parameters in the BW-type models of hysteresis. The plan of this Chapter is as follows: Section 7.1 presents the formulation of the optimization problem, which is later tested on the numerical experiments presented in Section 7.2. Section 7.3 gives concluding remarks and highlights about the proposed procedure to compute parameters of BW-type models of hysteresis.

## 7.1. BW-type models via MOBEAs

All the proposed methods for the identification of the parameters of the BW-type models listed in Chapter 6 try to reduce only the mean square error between the estimated displacements and the ones measured in the real structure. The proposed methodology herein uses multi-objective optimization techniques, so more than one objective function is formulated and a set of solutions is computed (pareto optimal front). This novel identification method simultaneously minimizes not only the dissimilarities between the estimated displacements and the real ones, but also between the dissipated energy by the structure and the one estimated using the mathematical model of the BW-type models.

### 7.1.1. Formulation of equations to minimize

We will employ the NSGA-II algorithm (see Section 5.2.1) in order to identify the parameters of the BW-type models. The idea is that an appropriate choice of the parameters must represent as well as possible the hysteretic behaviour of the structure. The parameters that will be tuned are represented by vector $\boldsymbol{p}$.
Suppose that we have a SDOF subjected to an external excitation (see Fig. **2-10**); the system is instrumented so the input signal (acceleration) $u(t)$ and the output signal (displacements) $x(t)$ are measured; a time-series with $l$ elements will be recorded. The BW-type model will be used to estimate the displacements of the structure, and those estimations will be compared with the experimental ones.
In order to assess the quality of the identification, the set of parameters $\boldsymbol{p}$ must minimize at the same time the following set of four objective functions:

1. The weighted error between the displacements measured in the laboratory $x(t)$ and the predicted displacements $\hat{x}\left(t|\boldsymbol{p}\right)$ with the BW-type model of hysteresis:

$$f_1\left(\boldsymbol{p}\right) = \sum_{i=1}^{l} \frac{\left|x\left(t_i\right) - \hat{x}\left(t_i|\boldsymbol{p}\right)\right|}{w\left(t_i\right)}, \tag{7-1}$$

here $w(t)$ is a weighting function used to normalize the displacements between $[-1, 1]$; such function is computed using an envelope that linearly interpolates between the peak values of the absolute value of the displacements; the weighting function is illustrated with the red dashed line in Fig. **7-1**. This procedure is carried out in order to give the same importance to small and large displacements.

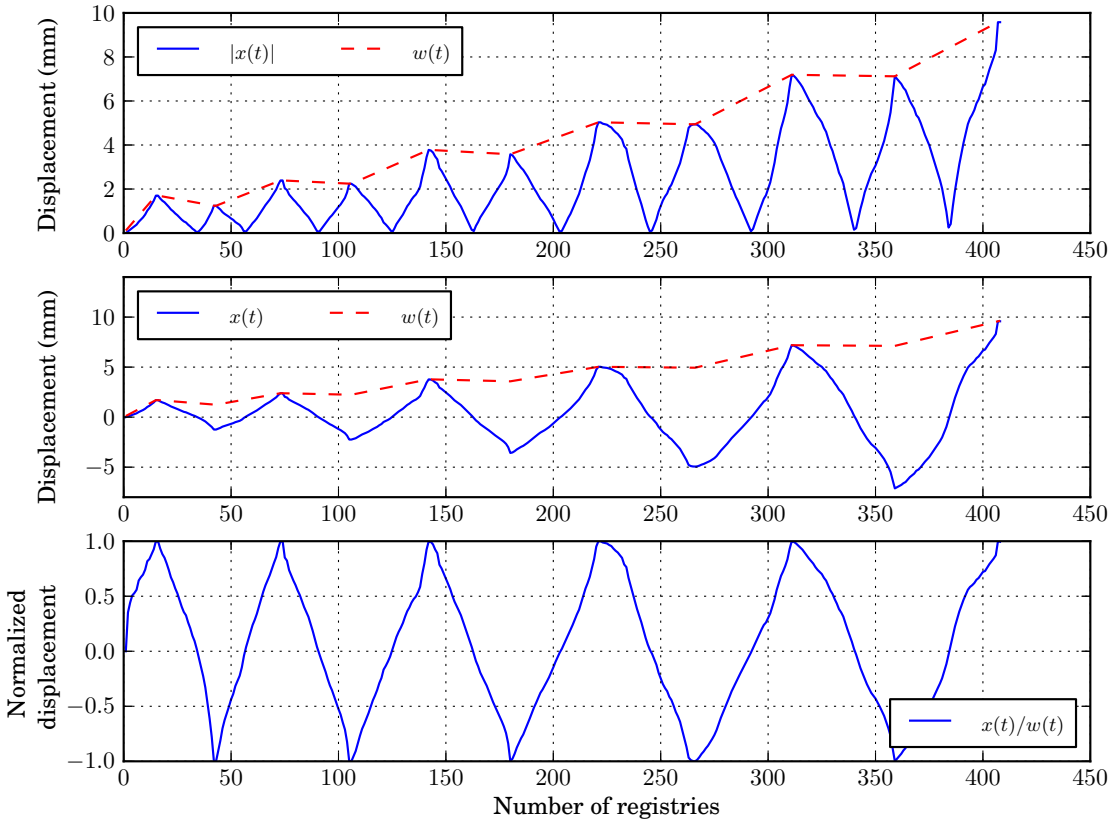



**Figure 7-1**.: Measured displacement and its normalization. This normalization is the one corresponding to the numerical experiment discussed in Section 7.2.3. The red dashed line forms the envelope $w(t)$ that is used to normalize the displacement

2. The maximum error between the displacements measured in the laboratory $x(t)$ and

the predicted displacements $\hat{x}\,(t|\boldsymbol{p})$:

$$f_2\,(\boldsymbol{p}) = \max_{1\leq i\leq l}\left\{\left|x\,(t_i) - \hat{x}\,(t_i|\boldsymbol{p})\,\right|\right\}.$$

The minimization of the objective functions $f_1$ and $f_2$ results in a better approximation of the displacements, so the hysteresis curve will be better adjusted.

3. The difference between the total dissipated energy per unit mass $\varepsilon_{tot}(t_l)$ calculated from the load-displacement records, and the total dissipated energy per unit mass $\hat{\varepsilon}_{tot}(t_l|\boldsymbol{p})$ estimated with the BW-type model at time $t_l$ for a set of parameters $\boldsymbol{p}$:

   $$f_3\,(\boldsymbol{p}) = \left|\varepsilon_{tot}(t_l) - \hat{\varepsilon}_{tot}(t_l|\boldsymbol{p})\right|;$$

   remember that $l$ is the number of elements in the time series. The total dissipated energy per unit mass is the sum of the hysteretic dissipated energy per unit mass $\varepsilon(t)$ (Eq. (2-10)) and the elastic dissipated energy per unit mass $\varepsilon_{el}(t)$, i.e.,

   $$\varepsilon_{tot}(t_l) = \varepsilon(t_l) + \varepsilon_{el}(t_l).$$

   Here $\varepsilon_{tot}(t_l)$ represents the area enclosed by the hysteresis curves, and $\varepsilon_{el}(t)$ is derived using the elastic restoring force $F^e(x(t)) := \alpha k_i x(t)$, as:

   $$\varepsilon_{el}(t) := \int_{x(0)}^{x(t)} \frac{F^e(x)}{m}\mathrm{d}x = \alpha\frac{k_i}{m}\int_0^t x(\tau)\dot{x}(\tau)\mathrm{d}\tau$$

   that is,

   $$\varepsilon_{el}(t) := \alpha\omega_0^2\int_0^t x(\tau)\dot{x}(\tau)\mathrm{d}\tau.$$

4. The maximum error between the dissipated energy per unit mass computed from records ($\varepsilon_{tot}(t)$) and the predicted dissipated energy per unit mass ($\hat{\varepsilon}_{tot}\,(t|\boldsymbol{p})$):

   $$f_4\,(\boldsymbol{p}) = \max_{1\leq i\leq l}\left\{\left|\varepsilon_{tot}\,(t_i) - \hat{\varepsilon}_{tot}\,(t_i|\boldsymbol{p})\,\right|\right\}. \tag{7-2}$$

   The minimization of the objective functions $f_3$ and $f_4$ ensures that each hysteresis cycle is well approximated, inasmuch as the total dissipated energy per unit mass is a cumulative measure, so a good approximation of the estimated hysteresis cycles with respect to the shape of the experimental hysteresis curves is expected.

To restrict the parameter space $\mathcal{D}$, two vectors $\boldsymbol{p}_{\min}$ and $\boldsymbol{p}_{\max}$ are defined, such that:

$$\boldsymbol{p}_{\min}(i) \leq \boldsymbol{p}(i) \leq \boldsymbol{p}_{\max}(i), \qquad 1 \leq i \leq q$$

so vector $\boldsymbol{p}$ is restricted to a region where the initial population can be generated, and also determines the feasible values that the parameters can take.

It has to be noted that some parameters of the differential equation (2-9) may diverge, and in such a case, the algorithm is forced to assign infinity to the response of the system, so this set of parameters will become infeasible in the parameter space $\mathcal{D}$. This step implicitly guarantees that inequalities typically used to ensure the Bounded Input-Bounded Output (BIBO) stability of the differential equation (2-9) such as $\beta + \gamma > 0$ and $\gamma - \beta < 0$ are satisfied.

### 7.1.2. Algorithm

For the identification process, the software developed by Deb [32] was employed, the source code can be found at Professor Deb website [32]. Using such source code as "main engine", the BW-type models of hysteresis and the four objective functions proposed above were implemented.
The program allows the user to work with binary-coded GAs or real-parameter GAs. For the simulations that will perform herein, real-parameter GAs are used. The unique stopping criterion that was imposed was the number of generations (iterations), so the algorithm can perform several runs and, given the nature of the problem, a good diversity of solutions is expected.

## 7.2. Numerical experiments

In order to evaluate the efficiency of the proposed methodology, several simulations were carried out using simulated data and real load-displacement records. The following cases were considered:

1. Bouc-Wen model of hysteresis (without degradation and pinching) using simulated data.
2. BWBN model using simulated data.
3. BWBN model using real load-displacements records.

In cases 1 and 2, we set $m = 500$ kg and $k_i = 6$ kN/mm, and the external excitation $u(t)$ was a record of 20 seconds of the signal $u(t) = t \sin(2\pi t)$ sampled each 0.02 seconds; in consequence, $l = 1001$ samples were employed (see Fig. **7-2**).
The exact response of the system was computed using the "true" parameters that appear in Tables **7-1** and **7-2**. Eq. (2-8) was evaluated using the fourth-order Runge-Kutta method. The proposed algorithm was applied in order to estimate those parameters. The values of $\boldsymbol{p}_{\min}$ and $\boldsymbol{p}_{\max}$ are also shown in Tables **7-1** and **7-2**.
Inasmuch as MOBEAs return a set of solutions (pareto optimal front), a criterion has to be imposed in order to select a unique solution among the whole population in the pareto optimal front; the one adopted here is the minimum Euclidean distance from the origin to

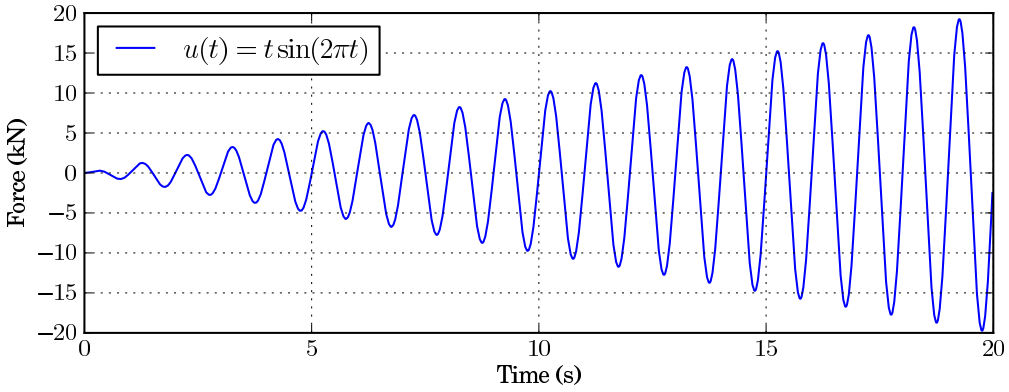



**Figure 7-2.**: External excitation used in the simulations.

a "normalized" pareto front that maps the objective functions to the range $[0, 1]$ in order to avoid scale differences:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}_k \in \mathcal{D}} \left\{ \sqrt{\sum_{i=1}^{4} \left( \frac{f_i\left(\boldsymbol{p}_k\right) - \min_{1\leq j\leq \mu}\left\{f_i\left(\boldsymbol{p}_j\right)\right\}}{\max_{1\leq j\leq \mu}\left\{f_i\left(\boldsymbol{p}_j\right)\right\} - \min_{1\leq j\leq \mu}\left\{f_i\left(\boldsymbol{p}_j\right)\right\}} \right)^2} \right\}. \tag{7-3}$$

### 7.2.1. Identification of the parameters of the Bouc-Wen model of hysteresis (without degradation and pinching)

In this case the degradation functions and the pinching function are set to one, that is $\nu(t) = A(t) = \eta(t) = h(t) = 1$, and the "true" parameters of the system were set up as shown in Table **7-1**, so the vector $\boldsymbol{p}$ in this case is given by:

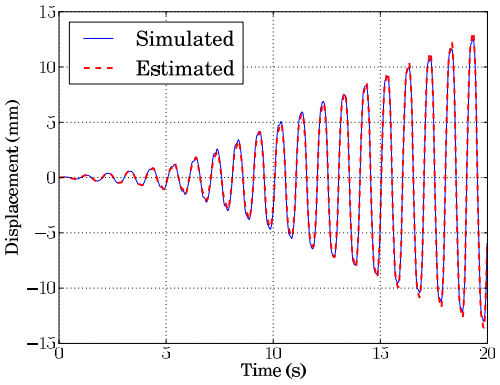
$$\boldsymbol{p} = [\xi, \alpha, \beta, \gamma, n]^T.$$

The algorithm was initialized with different seeds (in the random number generator). The results after 100 generations are shown in Table **7-1**.

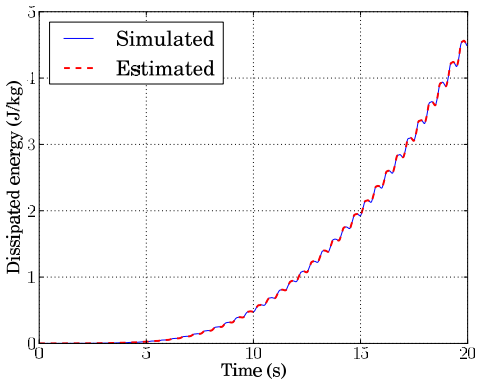
At first glance, it seems that the algorithm was unable to estimate the response of the system given the errors between the real parameters and the estimated ones, but looking at Fig. **7-3**, it can be seen, that in fact, the approximation was good.

Independent of the seed used to initialize the random number generator, the estimation of the parameters are very similar, and the values of $\alpha, \beta, \gamma$ and $n$ are relatively close to the real ones, but the estimated values of $\xi$ are larger than the real one; Charalampakis and Koumousis [24] observed the same phenomenon.

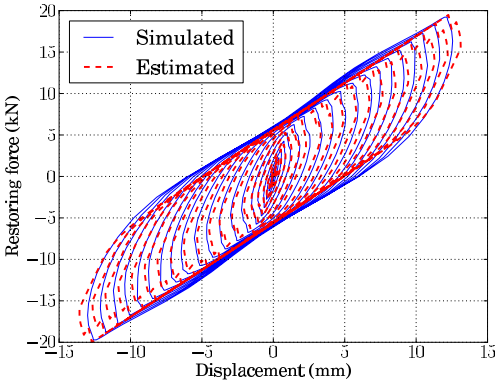
The evolution of the parameters $\beta$ and $\gamma$ of the Bouc-Wen model are shown in Fig. **7-4**. After 40 generations the parameters of the representative point according to Eq. (7-3) remain in the same range of values, so it can be considered that the algorithm has converged at this point of the simulation.




(a) Displacements




(b) Total dissipated energy per unit mass

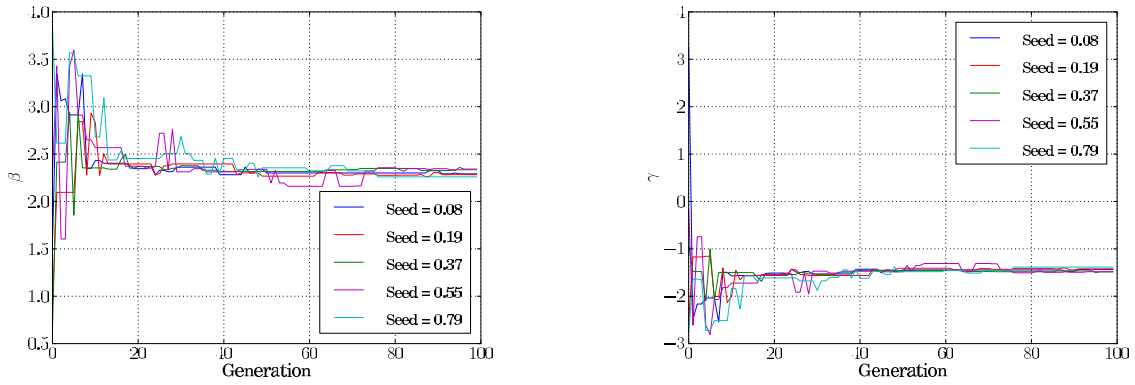



(c) Hysteresis cycles

**Figure 7-3**.: Simulated and estimated response of the system (Seed = 0.08) analyzed in Section 7.2.1

**Table 7-1**.: Identification results from simulated data analyzed in Section 7.2.1

| Param. | $\boldsymbol{p}_{\min}$ | True value | $\boldsymbol{p}_{\max}$ | Seeds used in the random number generator | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | 0.08 | 0.19 | 0.37 | 0.55 | 0.79 |
| $\xi$ | 0.01 | 0.05 | 0.20 | 0.1681 | 0.1664 | 0.1649 | 0.1677 | 0.1652 |
| $\alpha$ | 0.01 | 0.20 | 0.40 | 0.1885 | 0.1882 | 0.1873 | 0.1886 | 0.1863 |
| $\beta$ | 0.10 | 2.00 | 4.00 | 2.2916 | 2.2832 | 2.3417 | 2.3344 | 2.2595 |
| $\gamma$ | -4.00 | -1.00 | 4.00 | -1.4355 | -1.4230 | -1.4856 | -1.4826 | -1.3850 |
| $n$ | 1.00 | 1.20 | 5.00 | 1.0895 | 1.0792 | 1.1088 | 1.1029 | 1.0811 |
| $f_1(\boldsymbol{p})$ | - | 0 | - | 42.1208 | 41.4385 | 40.9654 | 42.4628 | 38.3287 |
| $f_2(\boldsymbol{p})$ | - | 0 | - | 0.6369 | 0.6593 | 0.7500 | 0.6514 | 0.8123 |
| $f_3(\boldsymbol{p})$ | - | 0 | - | 0.1059 | 0.1046 | 0.0637 | 0.0926 | 0.0809 |
| $f_4(\boldsymbol{p})$ | - | 0 | - | 0.1184 | 0.1165 | 0.0760 | 0.1057 | 0.0902 |




(a) $\beta$ (b) $\gamma$

**Figure 7-4**.: Evolution parameters Bouc-Wen model analyzed in Section 7.2.1

### 7.2.2. Identification of the parameters of the BWBN model of hysteresis

Now the full set of parameters used to describe hysteretic behaviour is considered for the identification process. The "true" parameters of the system were set up as shown in Table **7-2**, so the vector $\boldsymbol{p}$ in this case is given by:

$$\boldsymbol{p} = [\xi, \alpha, \beta, \gamma, n, \nu_0, \delta_\nu, A_0, \delta_A, \eta_0, \delta_\eta, p, \zeta_0, \psi_0, \delta_\psi, \lambda, q]^T .$$

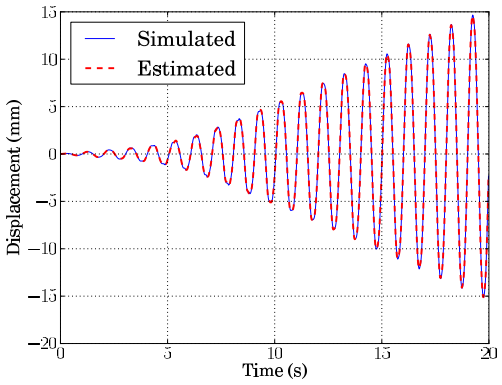
The number of generations in the NSGA-II was set to 400. Three different seeds were used to initialize the random number generator; the results are summarized in Table **7-2**.

**Table 7-2**.: Identification results from simulated data analyzed in Section 7.2.2

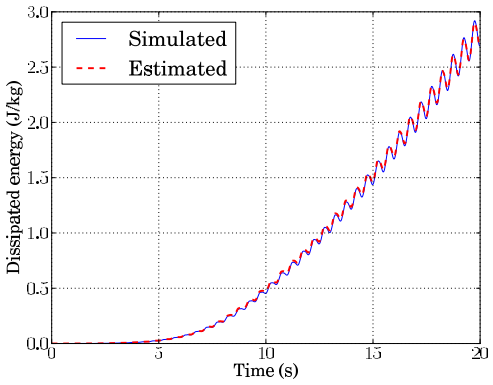
| Param. | $\boldsymbol{p}_{\min}$ | True value | $\boldsymbol{p}_{\max}$ | Seeds used in the random number generator | | |
|---|---|---|---|---|---|---|
| | | | | 0.66 | 0.78 | 0.92 |
| $\xi$ | 0.02 | 0.05 | 0.20 | 0.1516 | 0.1525 | 0.1514 |
| $\alpha$ | 0.01 | 0.20 | 0.30 | 0.1811 | 0.1843 | 0.1791 |
| $\beta$ | 0.10 | 2.00 | 4.00 | 3.7970 | 2.9324 | 2.7319 |
| $\gamma$ | -4.00 | -1.00 | 4.00 | -2.2009 | -1.9482 | -1.6778 |
| $n$ | 1.00 | 1.20 | 5.00 | 1.0819 | 1.2473 | 1.1786 |
| $\nu_0$ | 0.10 | 1.20 | 3.00 | 1.8447 | 1.3742 | 1.2747 |
| $\delta_\nu$ | -2.00 | 0.40 | 4.00 | 0.5990 | -0.0203 | -0.0485 |
| $A_0$ | 0.50 | 1.10 | 3.00 | 2.8818 | 1.2882 | 1.2849 |
| $\delta_A$ | -2.00 | 0.10 | 3.00 | -0.4595 | 0.1749 | 0.1264 |
| $\eta_0$ | 0.50 | 1.30 | 4.00 | 3.1990 | 1.4781 | 1.4693 |
| $\delta_\eta$ | -3.00 | 0.50 | 4.00 | 2.7462 | 1.4963 | 1.8161 |
| $p$ | 0.00 | 2.00 | 9.00 | 1.1477 | 5.6160 | 2.0806 |
| $\zeta_0$ | -8.00 | 1.00 | 8.00 | 0.8526 | 0.5605 | 0.5668 |
| $\psi_0$ | -4.00 | 0.50 | 4.00 | -0.1886 | 0.3436 | -0.3474 |
| $\delta_\psi$ | -4.00 | 0.60 | 4.00 | 2.8794 | -1.9170 | -2.2260 |
| $\lambda$ | -4.00 | 0.90 | 4.00 | 1.3055 | -1.6739 | 0.5382 |
| $q$ | -4.00 | 1.00 | 4.00 | 3.9055 | 1.5380 | 2.0701 |
| $f_1(\boldsymbol{p})$ | - | 0 | - | 20.1579 | 22.1891 | 21.4904 |
| $f_2(\boldsymbol{p})$ | - | 0 | - | 0.2513 | 0.2517 | 0.2469 |
| $f_3(\boldsymbol{p})$ | - | 0 | - | 0.0031 | 0.0021 | 0.0038 |
| $f_4(\boldsymbol{p})$ | - | 0 | - | 0.0224 | 0.0248 | 0.0220 |

It can be seen that the parameters did not converge to the real values; however, the approximation with the estimated parameters is good enough for simulation purposes. Fig. **7-5**

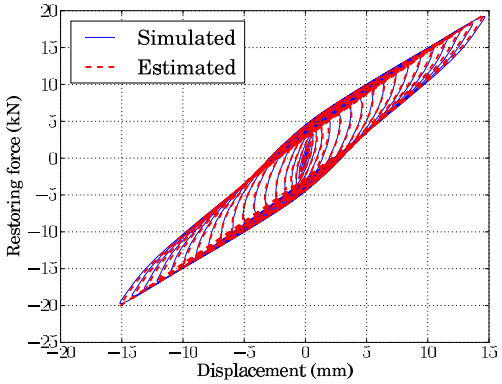
and **7-6** show results drawn from two different simulations; as it can be seen, the estimation with both set of parameters are very similar.




(a) Displacements



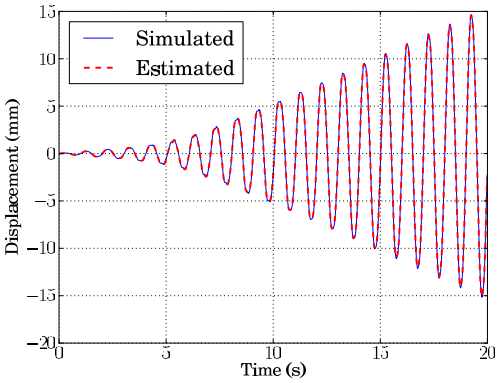

(b) Total dissipated energy per unit mass



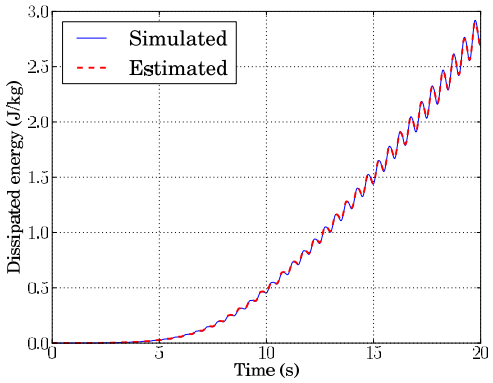

(c) Hysteresis cycles

**Figure 7-5**.: Simulated and estimated response of the system (Seed = 0.66) analyzed in Section 7.2.2

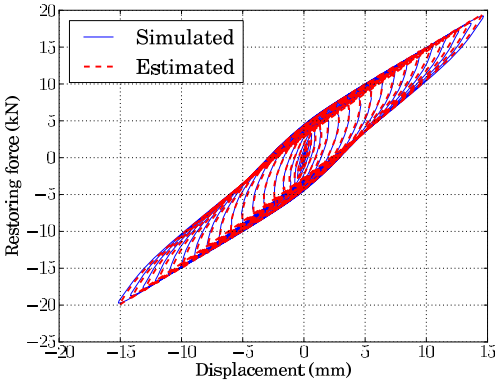
The variation of the parameters $\beta$ and $\gamma$ of the BWBN model is shown in Fig. **7-7**, After 300 generations the parameters stay in the same range of values.
In order to test whether the parameters in Table **7-2** represent well the hysteretic response of the system, we simulated its response to a different input excitation than the one used in the system identification. We used as input excitation the 1994 Northridge earthquake (signal retrieved from [22]), so that $u(t) = -\ddot{x}_g(t)$, i.e., an acceleration at the base of the SDOF system under study.
Fig. **7-8** shows the resulting displacements using the real parameters and the estimated ones using the NSGA-II. It can be seen that the displacements are similar in shape, and the amplitude varies slightly among solutions. Fig. **7-8** displays the frequency responses computed from the displacements, as well; one can see that the power and the frequency components are very similar for all sets of parameters. Even if the parameters differ significantly from the




(a) Displacements

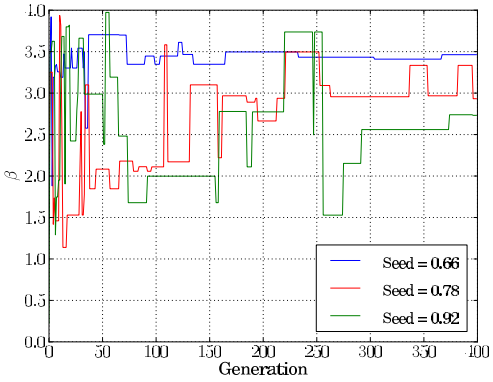



(b) Total dissipated energy per unit mass

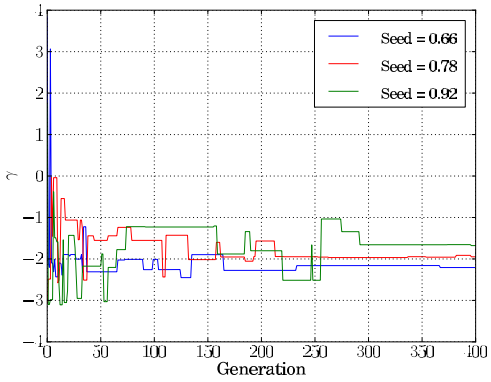



(c) Hysteresis cycles

**Figure 7-6**.: Simulated and estimated response of the system (Seed = 0.92) analyzed in Section 7.2.2




(a) $\beta$




(b) $\gamma$

**Figure 7-7**.: Evolution parameters BWBN model analyzed in Section 7.2.2

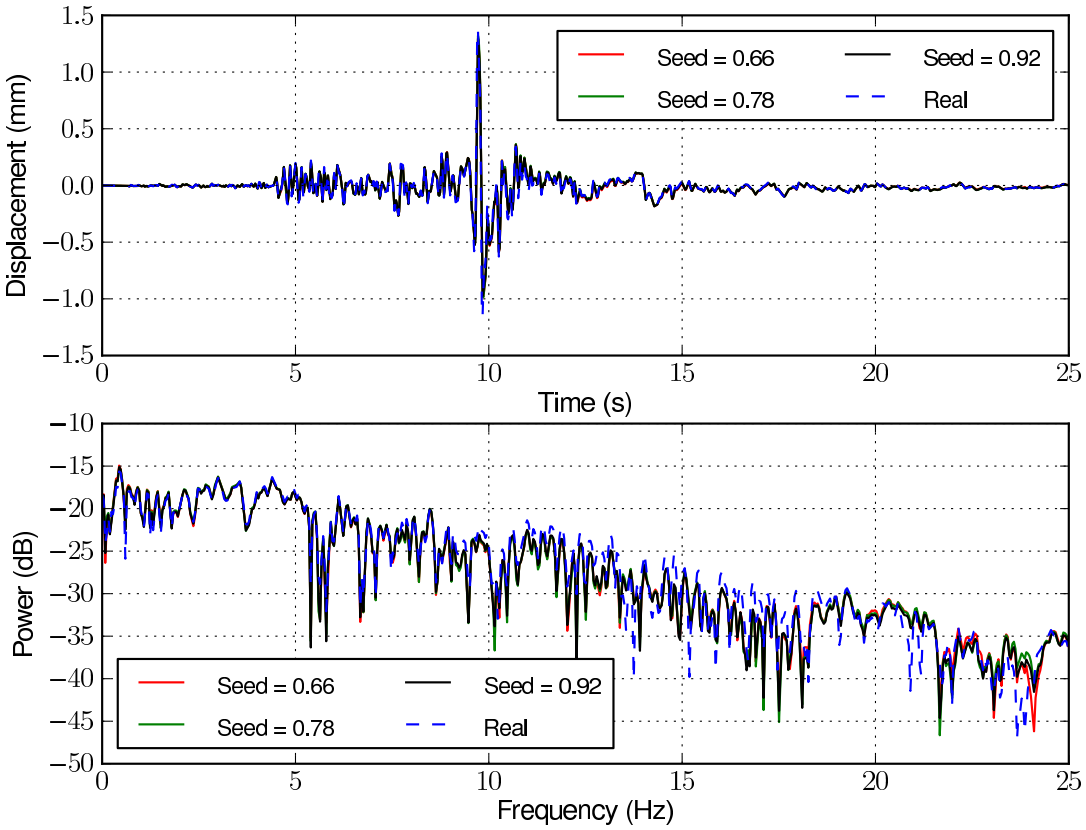



**Figure 7-8.**: Estimated displacements and spectrum for the 1994 Northridge earthquake and the BWBN model analyzed in Section 7.2.2

real ones, if the responses of the system computed with the estimations are similar to those obtained via the real parameters, then the frequency response will just change its magnitude but not its frequency components.

### 7.2.3. Identification using force-displacement records measured at the laboratory

In the previous section, the identification of the parameters of the BW-type models of hysteresis from simulated data was performed. The results showed that for a given system, the hysteretic behaviour can be represented fairly well by different set of parameters, so it can be concluded that the BW-type models of hysteresis are multi-modal functions, where several minima exist; this renders the process of finding the real parameters of the models an almost impossible task, but from a practical standpoint, the estimated parameters through the proposed methodology are useful, because, without regard of the applied external excitation, the BW-type model of hysteresis, provided with the estimated parameters, can recreate the responses of the systems with an acceptable precision.

Now the algorithm will be applied to force-displacements records measured at the laboratory, and it will be seen wether or not the same conclusion can be drawn from these data. The registries used herein were obtained from cyclic tests carried out in ferrocement models tested in [13] and [12]. The model is a full scale wall subjected to an increasing cyclic load; the wall had 1 m width, 2 m height and 0.02 m thick. The wall was anchored to the foundation beam illustrated in Fig. **7-9**; this beam was fixed to the floor. Lateral bracing was used in order to avoid lateral displacements of the wall, so that only in-plane displacements were allowed. The displacements induced by the actuator were measured with LVDTs located in the top of the wall. For the construction of the thin ferrocement walls, mortar and wire mesh were used. The mortar was made of Portland cement and sand in a weight relation of 1:2, and the water/cement ratio was 0.4. The resistance of the mortar after 28 days was $f'_c = 33$ MPa. The mass and initial stiffness of the wall were determined as $m = 456$ kg, $k_i = 6.2684$ kN/mm. This mass corresponds to the ferrocement wall and to the roof.

The NSGA-II was set up with the following values: parent population size $\mu = 10000$, offspring population size $\theta = 10000$, 100 generations, real-parameter codification with simulated binary crossover and polynomial mutation, with crossover distribution index $\eta_c = 15$ and mutation distribution index $\eta_m = 20$. The BWBN model of hysteresis is adopted for the identification process. The results for three different simulations started with different seeds in the random number generator are reported in Table **7-3**

It can be seen from Fig. **7-10** that the estimated response of the system is in good agreement with the measured one. Just as in the simulated cases, several runs started with different seeds generate different sets of parameters that are able to reproduce the response of the system fairly well.

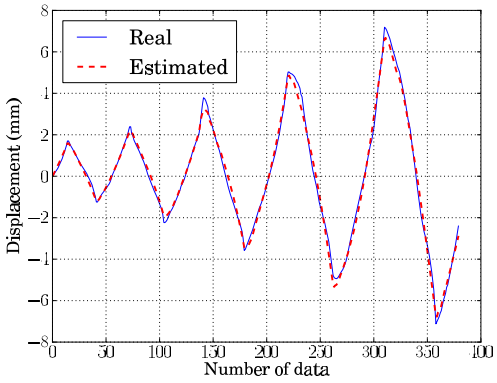
The estimated parameters shown in Table **7-3** will be used to predict the displacements

**Figure 7-9**.: Ferrocement wall analyzed in Section 7.2.3

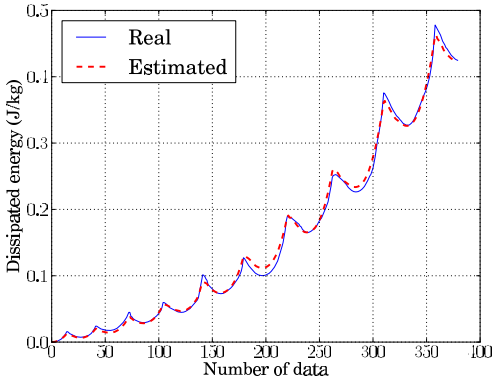
**Table 7-3**.: Identification results from the model analyzed in Section 7.2.3

| Parameter | $\boldsymbol{p}_{\min}$ | $\boldsymbol{p}_{\max}$ | Seeds used in the random number generator | | |
|---|---|---|---|---|---|
| | | | 0.09 | 0.35 | 0.81 |
| $\xi$ | 0.02 | 0.20 | 0.1702 | 0.1977 | 0.1813 |
| $\alpha$ | 0.10 | 0.30 | 0.1988 | 0.2183 | 0.2128 |
| $\beta$ | 0.10 | 4.00 | 1.5001 | 3.0250 | 3.1422 |
| $\gamma$ | -4.00 | 4.00 | -1.1510 | -1.3156 | -0.8267 |
| $n$ | 1.00 | 5.00 | 1.8813 | 1.5173 | 2.1062 |
| $\nu_0$ | 0.10 | 3.00 | 0.7896 | 0.4092 | 0.3075 |
| $\delta_\nu$ | -2.00 | 4.00 | 2.7525 | 3.2397 | 3.0769 |
| $A_0$ | 0.50 | 3.00 | 1.9241 | 2.4929 | 2.2163 |
| $\delta_A$ | -2.00 | 3.00 | 1.5240 | 0.8155 | 0.7824 |
| $\eta_0$ | 0.50 | 4.00 | 2.5757 | 3.4873 | 3.0770 |
| $\delta_\eta$ | -3.00 | 4.00 | 3.0275 | -0.8492 | 3.0670 |
| $p$ | 0.00 | 10.00 | 7.8931 | 9.7076 | 4.9373 |
| $\zeta_0$ | -8.00 | 8.00 | 1.3344 | 1.0962 | 2.2306 |
| $\psi_0$ | -4.00 | 4.00 | -3.1148 | 1.2478 | -3.4286 |
| $\delta_\psi$ | -4.00 | 4.00 | 0.4828 | -3.6339 | 1.3153 |
| $\lambda$ | -4.00 | 4.00 | -1.4976 | -3.2035 | -1.8824 |
| $q$ | -4.00 | 4.00 | 0.9635 | 1.5662 | 3.0920 |
| $f_1$ | - | - | 20.2848 | 21.2538 | 22.8303 |
| $f_2$ | - | - | 0.6964 | 0.6914 | 0.7499 |
| $f_3$ | - | - | 0.0014 | 0.0014 | 0.0001 |
| $f_4$ | - | - | 0.0164 | 0.0203 | 0.0153 |

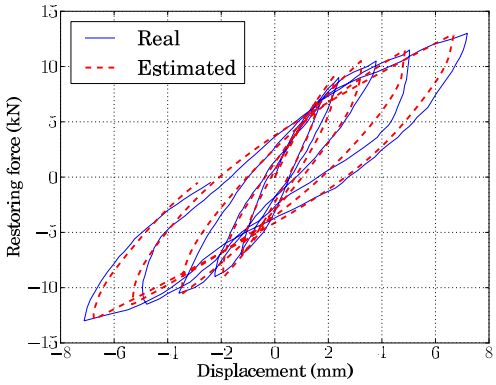
suffered by the system when it is subjected to a random excitation. The 1994 Northridge earthquake is used as external excitation $u(t) = -\ddot{x}_g(t)$. The resulting displacements and the frequency response using these parameters are show in Fig. **7-8**. It can be seen that the estimated displacements and the frequency components are similar for the three set of




(a) Displacements




(b) Total dissipated energy per unit mass




(c) Hysteresis cycles

**Figure 7-10**.: Experimental and estimated response of the system (Seed = 0.09) analyzed in Section 7.2.3

parameters. The displacements are very small for such strong excitation. This shows the efficiency of the ferrocement as an alternative low cost construction system.

## 7.3. Final remarks and highlights

A novel methodology to estimate the parameters of BW-type models of hysteresis has been proposed. The procedure uses the NSGA-II, which is an elitist multi-objective optimization algorithm. The identification method has shown its ability to find sets of parameters that are capable of reproducing the response of the system with precision.

The numerical experiments give a glimpse of the multi-modality of the BWBN model of hysteresis. The NSGA-II procedure is able to minimize the proposed objective functions and approximate the hysteresis curves, and that is achieved using several local minima that are able to estimate the response of the system. Even though it was not possible to estimate the exact values of the parameters related with the model, the response of the system, both in

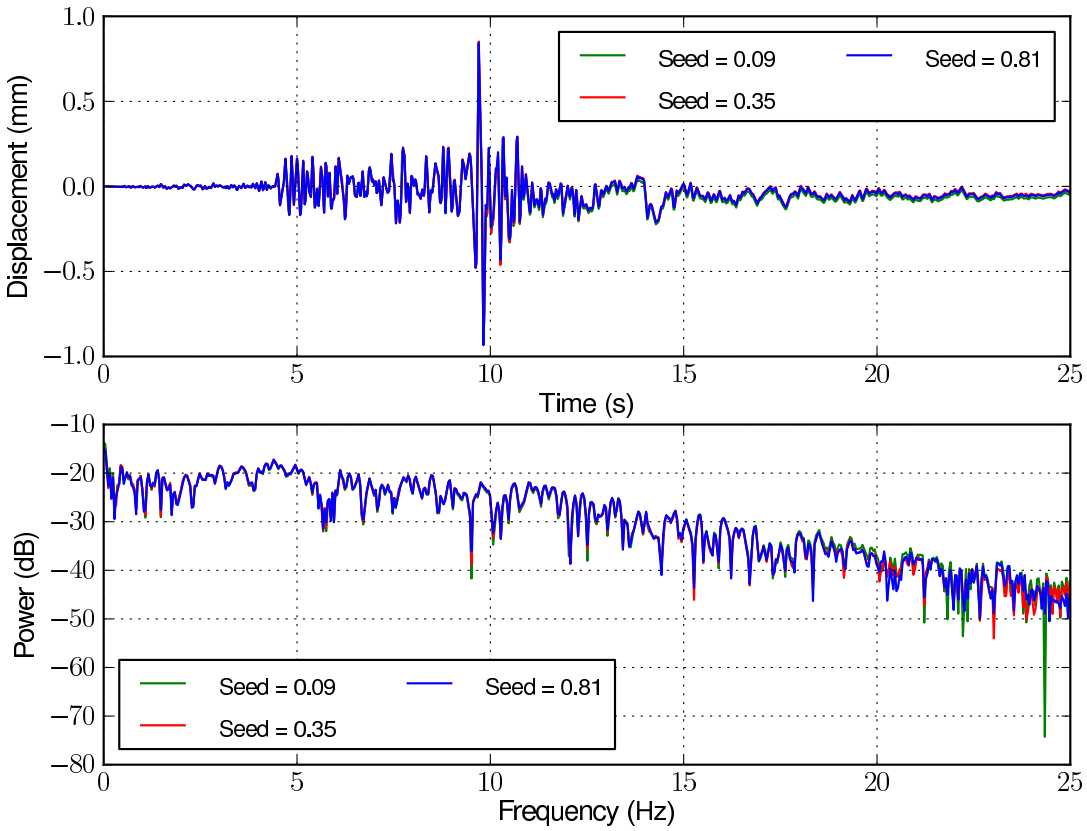



**Figure 7-11**.: Estimated displacements and spectrum for the 1994 Northridge earthquake and the BWBN model analyzed in Section 7.2.3

time and frequency, evaluated with the estimated parameters is good enough for simulation purposes. The multi-modality of the objective functions may shed lights on the fact that the BWBN model of hysteresis is "over parametrized".

Probably the cause that we did not find the global optimum in the optimization lies on the fact that we are optimizing in a relatively large parameter space $\mathcal{D}$. Aside of the multi-modality and strong nonlinearity of the BW-type models, the high dimension of $\mathcal{D}$ given by the number of parameters involved in the BWBN model of hysteresis, renders the sampling process a difficult task due to the so-called "curse of dimensionality" [14], which basically states that the higher the dimension of the space, the larger the number of samples needed to make an appropriate sampling. For instance, consider the unit interval $[0, 1] \in \mathbb{R}$, and 100 points evenly distributed in such interval; that distribution is enough to characterize the interval. Now consider the same 100 points distributed in the unit square in $\mathbb{R}^2$; these points will have a sparse distribution in such space, and they do not cover the whole space, in fact one point will cover the area of a square with side $0.01^{1/2} = 0.1$; the same 100 points in a unit

cube in $\mathbb{R}^3$ will represent each a cube with side $0.01^{1/3} = 0.215$. For a space of dimension $q$, the edge of the hyper-cube will be $0.01^{1/q}$; since in the case of the BWBN we are dealing with a parameter space $\mathcal{D} \subseteq \mathbb{R}^q$, of $q = 17$ dimensions, each of the 100 points will represent a hyper-cube with side $0.01^{1/17} = 0.763$. Therefore, if the dimension of the space increases, the 100 points considered will not be enough to make an appropriate sampling in the space, this is an example of the curse of dimensionality. This shows that it is very complicated to achieve the global minimum in high dimensions, and that it is advisable to increase the number of parents instead of the number of generations in order to try to overcome this problem.

The proposed methodology can be extended to handle more objective functions that will help to reduce the discrepancy between the real data and the estimated one. Furthermore, other hysteretic models such as biaxial models or asymmetric hysteresis should be analyzed using the proposed procedure in order to evaluate the efficiency of the algorithm when handling other nonlinear hysteretic models.

# 8. Conclusions

A novel methodology to estimate the parameters of BW-type models of hysteresis has been proposed. The procedure uses the NSGA-II, which is an elitist multi-objective optimization algorithm. The identification method has shown its ability to find sets of parameters that are capable to reproduce the response of the system with precision.
The numerical experiments give a glimpse of the multi-modality of the BWBN model of hysteresis. The NSGA-II procedure is able to minimize the proposed objective functions and approximate the hysteresis curves, and that is achieved using several local minima that are able to estimate the response of the system. Even though it was not possible to estimate the exact values of the parameters related with the model, the response of the system evaluated with the estimated parameters is good enough for simulation purposes.
The proposed methodology can be extended to handle more objective functions that will help to reduce the discrepancy between the real data and the estimated one. Furthermore, other hysteretic models such as biaxial models or asymmetric hysteresis should be analyzed using the proposed procedure in order to evaluate the efficiency of the algorithm when handling other nonlinear hysteretic models.
Finally, given the multi-modality of the BW-type models, other techniques should be reviewed. For instance, the filters of the Kalman family, have been used in the identification of hysteretic structural systems, and have shown their ability to converge to a set of parameters that are able to characterize the hysteretic behaviour of the system, but according to the observed so far, it is possible to infer that those techniques are prone to find local minima instead of the global optimum inasmuch as their performance depends on the selection of an initial state and initial covariance matrix.
There are additional points that can be targeted as future work:

- We used a weighting function to normalize the displacements in the first objective function to minimize (Eq. (7-1)). It would be advisable to do the same normalization with the fourth objective function (Eq. (7-2)), so small and large quantities of the dissipated energy will have the same importance.

- We found that the Unscented Kalman Filter (UKF) is not able to estimate the parameters of BW-type models of hysteresis. It is recommended to make a review of those techniques based on nonlinear filtering, and try to apply restrictions to the algorithms in order to improve the search capabilities of them. There is a methodology proposed by Kandepu *et al.* [58], but we think this methodology is rather simple and

may not work properly. We suggest to make a review of the Unscented transform and try to modify the Gaussian quadrature used in such methodology (Gauss-Hermite quadrature), so a proper relocation of the sigma points due to the defined constraints (physical constraints) is done.

- There are other filtering techniques that may seem promising for the identification of hysteretic structural systems. For example, Särkkä [82] developed an algorithm known as the *Kalman-Bucy* filter, that is used for recursive estimation of states of systems that are modelled as discretely observed stochastic differential equations, so it can be used for continuous-discrete filtering problems, where the classical Kalman filter cannot be used, because this new technique is able to handle stochastic differential equations with nonlinear and non-Gaussian components.

- It was shown that the BW-type models of hysteresis are overparameterized. Researchers must find an alternative model to the BW formulation that removes such overparameterization justifying the parameters of the new model from a mathematical viewpoint, and verify experimentally that the new model do not suffer from overfitting. Such model should be the simplest possible from the viewpoint of Occam's Razor, but at the same time, must have the necessary generality to represent all the possible hysteresis cycles that may arise in the laboratory. A desirable property that the new model should possess is that it does not contain many local minima, but a unique minimum.

# Part IV.

# Appendices

# A. Test functions for optimization

Test functions (known as artificial landscapes) are useful to evaluate characteristics of optimization algorithms, such as:

- Velocity of convergence.
- Precision.
- Robustness.
- General performance.

In this Chapter some test functions are presented with the aim of giving an idea about the different situations that optimization algorithms have to face when coping with these kind of problems. In the first part, some objective functions for single–objective optimization cases are presented. In the second part, test functions with their respective Pareto fronts for multi–objective optimization problems (MOP) are given.
The artificial landscapes presented herein for single–objective optimization problems are taken from [8], [47] and from Rody Oldenhuis software [72]. Given the amount of problems (55 in total), just a few are presented here. The complete list of test functions is found on the CD accompanying this document, and also on Internet [74].
The test functions used to evaluate the algorithms for MOP were taken from [33], [18] and [17]. You can download the software developed by Deb [32], which implements the NSGA-II procedure with GAs, or the program posted on Internet [75], which implements the NSGA-II procedure with ES.
Just a general form of the equation, a plot of the objective function, boundaries of the object variables and the coordinates of global minima are given herein.

# A.1. Test functions for single–objective optimization problems

- **Sphere function:** (Fig. **A-1**(a))

  $$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2.$$

  Minimum: $f\left(\underbrace{0, 0, \ldots, 0}_{n \text{ times}}\right) = 0$, for $-\infty \leq x_i \leq \infty$, $1 \leq i \leq n$.

- **Rosenbrock function:** (Fig. **A-1**(b))

  $$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right].$$

  $$\text{Minimum:} \begin{cases} n = 2 & \rightarrow \quad f(1,1) = 0. \\ n = 3 & \rightarrow \quad f(1,1,1) = 0. \\ n > 3 & \rightarrow \quad f\left(-1, \underbrace{1, 1, \ldots, 1}_{(n-1) \text{ times}}\right) = 0. \end{cases}$$

  for $-\infty \leq x_i \leq \infty$, $1 \leq i \leq n$.

- **Beale's function:** (Fig. **A-1**(c))

  $$f(x,y) = (1.5 - x + xy)^2 + \left(2.25 - x + xy^2\right)^2 + \left(2.625 - x + xy^3\right)^2.$$

  Minimum: $f(3, 0.5) = 0$, for $-4.5 \leq x, y \leq 4.5$.

- **Goldstein Price function:** (Fig. **A-1**(d))

  $$f(x,y) = \left(1 + (x + y + 1)^2 \left(19 - 14x + 3x^2 - 14y + 6xy + 3y^2\right)\right) \cdot \left(30 + (2x - 3y)^2 \left(18 - 32x + 12x^2 + 48y - 36xy + 27y^2\right)\right).$$

  Minimum: $f(0, -1) = 3$, for $-2 \leq x, y \leq 2$.

- **Booth's function:** (Fig. **A-1**(e))

  $$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2.$$

  Minimum: $f(1, 3) = 0$, for $-10 \leq x, y \leq 10$.

- **Bukin function N. 6:** (Fig. **A-1**(f))

  $$f(x,y) = 100\sqrt{|y - 0.01x^2|} + 0.01\,|x + 10|\,.$$

  Minimum: $f(-10,1) = 0$, for $-15 \leq x \leq -5$, $-3 \leq y \leq 3$.

- **Ackley's function:** (Fig. **A-1**(g))

  $$f(x,y) = -20\exp\left(-0.2\sqrt{0.5\,(x^2+y^2)}\right) - \exp\left(0.5\left(\cos(2\pi x) + \cos(2\pi y)\right)\right) + 20 + e.$$

  Minimum: $f(0,0) = 0$, for $-5 \leq x, y \leq 5$.

- **Matyas function:** (Fig. **A-1**(h))

  $$f(x,y) = 0.26\left(x^2+y^2\right) - 0.48xy.$$

  Minimum: $f(0,0) = 0$, for $-10 \leq x, y \leq 10$.

- **Lévy function N. 13:** (Fig. **A-1**(i))

  $$f(x,y) = \sin^2(3\pi x) + (x-1)^2\left(1 + \sin^2(3\pi y)\right) + (y-1)^2\left(1 + \sin^2(2\pi y)\right).$$

  Minimum: $f(1,1) = 0$, for $-10 \leq x, y \leq 10$.

- **Three Hump Camel function:** (Fig. **A-1**(j))

  $$f(x,y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2.$$

  Minimum: $f(0,0) = 0$, for $-5 \leq x, y \leq 5$.

- **Easom function:** (Fig. **A-1**(k))

  $$f(x,y) = -\cos(x)\cos(y)\exp\left(-\left((x-\pi)^2 + (y-\pi)^2\right)\right).$$

  Minimum: $f(\pi,\pi) = -1$, for $-100 \leq x, y \leq 100$.

- **Cross–in–tray function:** (Fig. **A-1**(l))

  $$f(x,y) = -0.0001\left(\left|\sin(x)\sin(y)\exp\left(\left|100 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}.$$

$$\text{Minima:} \begin{cases} f\left(1.34941, -1.34941\right) & = -2.06261 \\ f\left(1.34941, 1.34941\right) & = -2.06261 \\ f\left(-1.34941, 1.34941\right) & = -2.06261 \\ f\left(-1.34941, -1.34941\right) & = -2.06261 \end{cases}$$

for $-10 \leq x, y \leq 10$.

- **Eggholder function:** (Fig. **A-1**(m))

$$f\left(x, y\right) = -\left(y + 47\right) \sin\left(\sqrt{\left|y + \frac{x}{2} + 47\right|}\right) - x \sin\left(\sqrt{\left|x - \left(y + 47\right)\right|}\right).$$

Minimum: $f\left(512, 404.2319\right) = -959.6407$, for $-512 \leq x, y \leq 512$.

- **Hölder table function:** (Fig. **A-1**(n))

$$f\left(x, y\right) = -\left|\sin\left(x\right) \cos\left(y\right) \exp\left(\left|1 - \frac{\sqrt{x^2 + y^2}}{\pi}\right|\right)\right|.$$

$$\text{Minima:} \begin{cases} f\left(8.05502, 9.66459\right) & = -19.2085 \\ f\left(8.05502, 9.66459\right) & = -19.2085 \\ f\left(-8.05502, -9.66459\right) & = -19.2085 \\ f\left(-8.05502, -9.66459\right) & = -19.2085 \end{cases}$$

for $-10 \leq x, y \leq 10$.

- **McCormick function:** (Fig. **A-1**(o))

$$f\left(x, y\right) = \sin\left(x + y\right) + \left(x - y\right)^2 - 1.5x + 2.5y + 1.$$

Minimum: $f\left(-0.54719, -1.54719\right) = -1.9133$, for $-1.5 \leq x \leq 4$, $-3 \leq y \leq 4$.

- **Schaffer function N. 2:** (Fig. **A-1**(p))

$$f\left(x, y\right) = 0.5 + \frac{\sin^2\left(x^2 - y^2\right) - 0.5}{\left(1 + 0.001\left(x^2 + y^2\right)\right)^2}.$$

Minimum: $f\left(0, 0\right) = 0$, for $-100 \leq x, y \leq 100$.

- **Schaffer function N. 4:** (Fig. **A-1**(q))

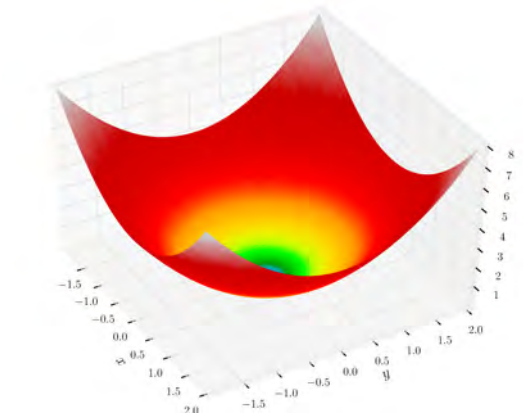
  $$f(x,y) = 0.5 + \frac{\cos\left(\sin\left(|x^2 - y^2|\right)\right) - 0.5}{\left(1 + 0.001\left(x^2 + y^2\right)\right)^2}.$$

  Minimum: $f(0, 1.25313) = 0.292579$, for $-100 \leq x, y \leq 100$.

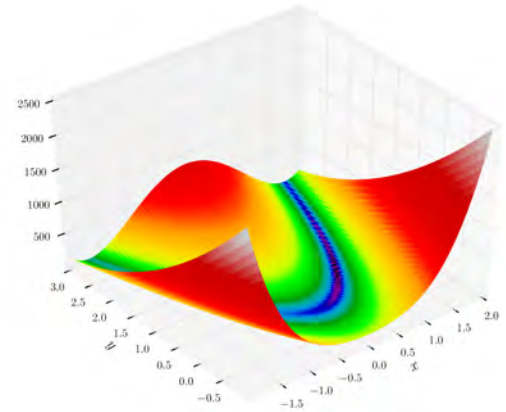
- **Styblinski–Tang function:** (Fig. **A-1**(r))

  $$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x}{2}.$$

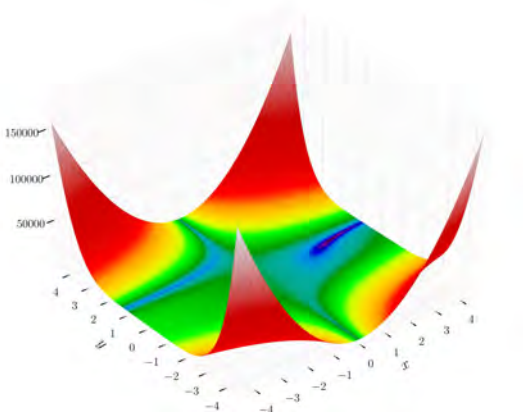
  Minimum: $f\left(\underbrace{-2.903534, \ldots, -2.903534}_{(n) \text{ times}}\right) = -39.16599n$, for $-5 \leq x_i \leq 5$, $1 \leq i \leq n$.



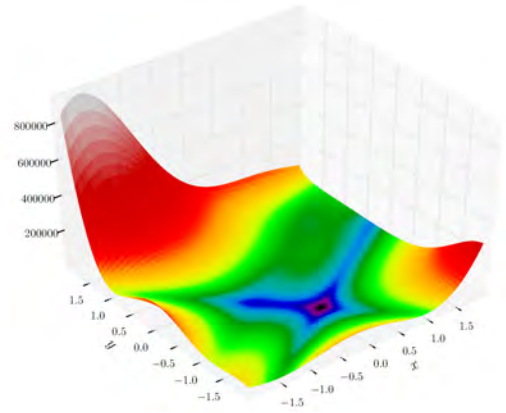
(a) Sphere function



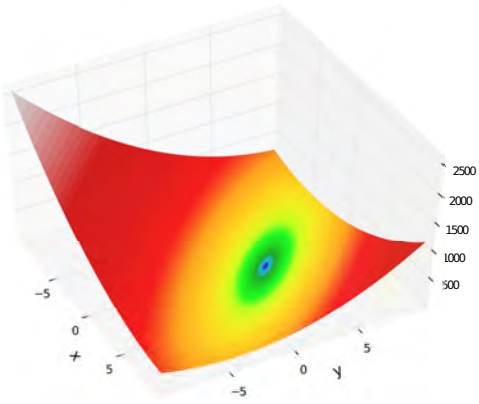
(b) Rosenbrock's function (2 variables)



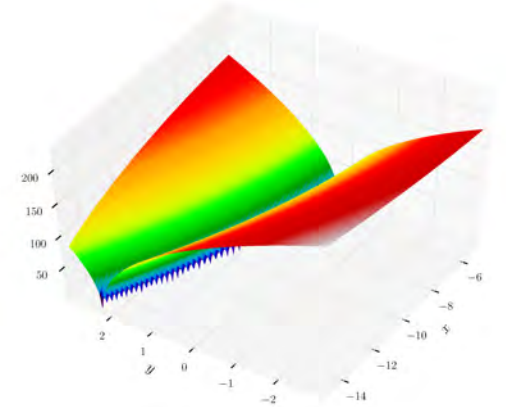
(c) Beale's function



(d) Goldstein Price function

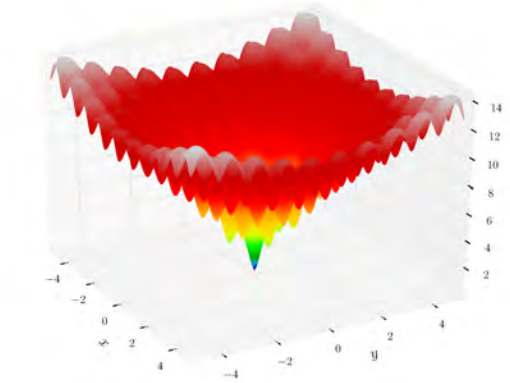


(e) Booth's function

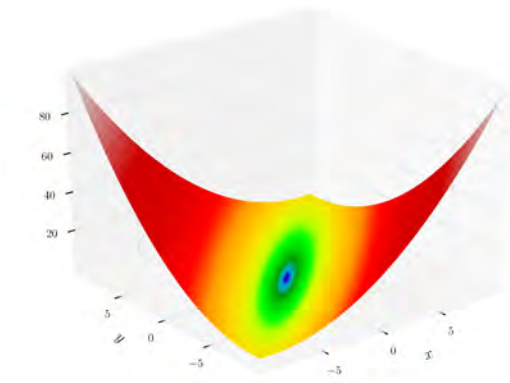


(f) Bukin function N. 6

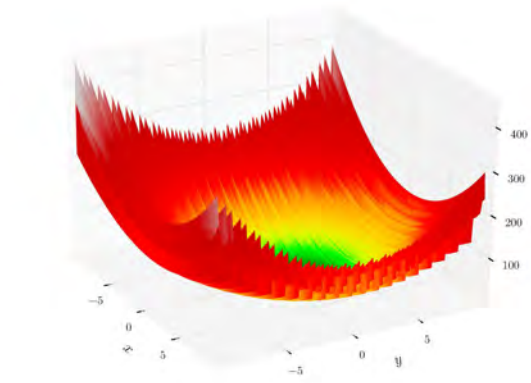
**Figure A-1**.: Test functions for single–objective optimization



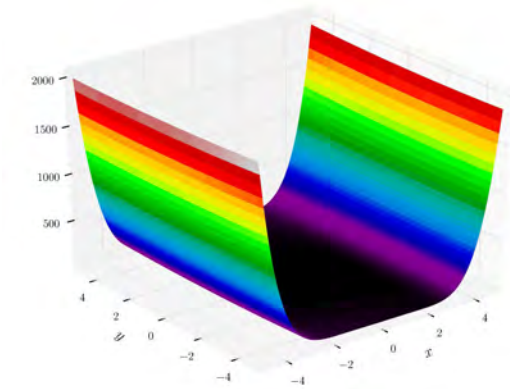
(g) Ackley's function



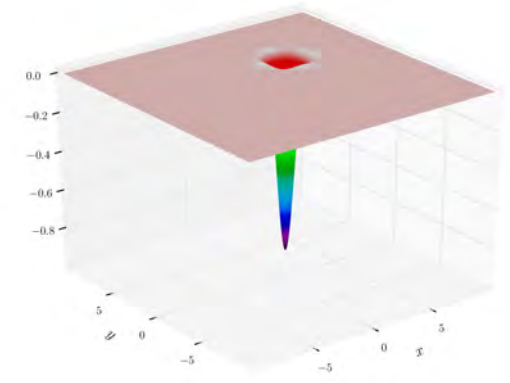
(h) Matyas function



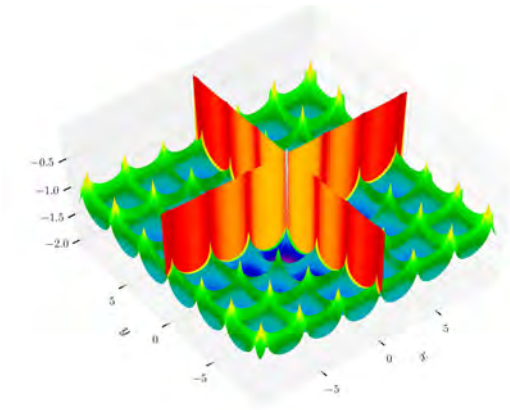
(i) Lévy function N. 13



(j) Three Hump Camel function

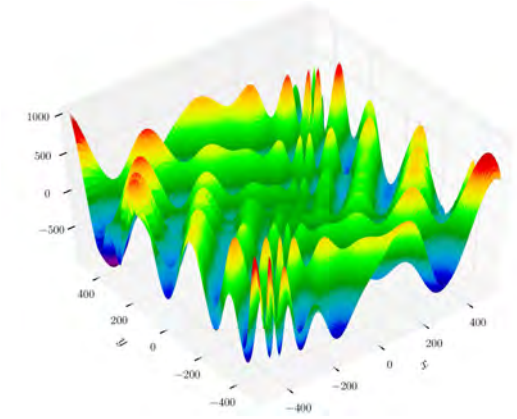


(k) Easom function

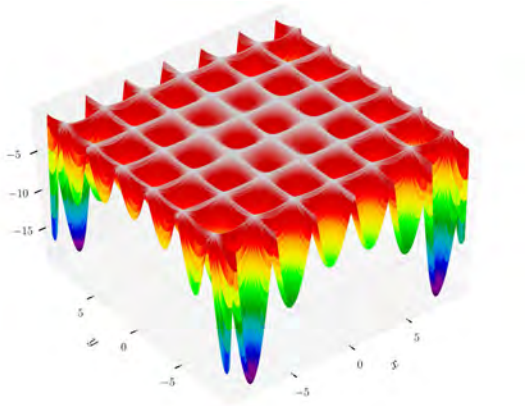


(l) Cross–in–tray function

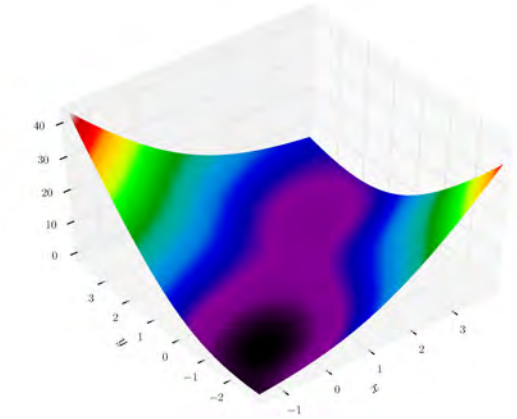
**Figure A-1**.: Test functions for single–objective optimization



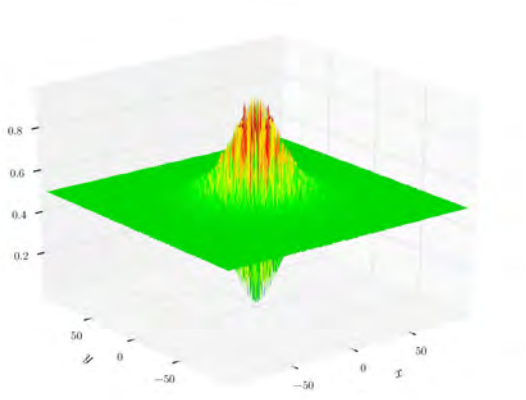
(m) Eggholder function



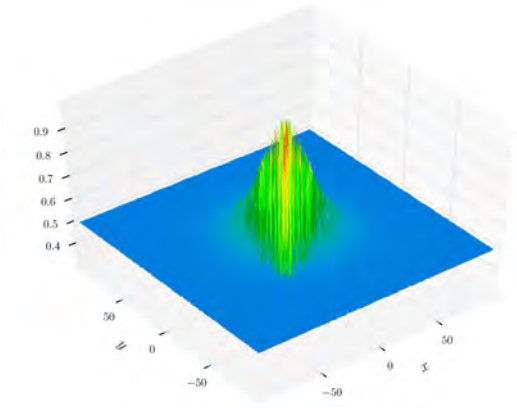
(n) Hölder table function



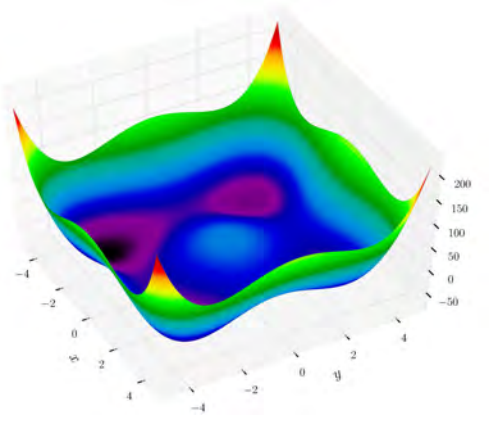
(o) McCormick function



(p) Schaffer function N. 2



(q) Schaffer function N. 4



(r) Styblinski–Tang function (2 variables)

**Figure A-1**.: Test functions for single–objective optimization

## A.2. Test functions for multi–objective optimization problems

- **Binh and Korn function:** (Fig. **A-2**(a))

$$\text{Minimize: } \begin{cases} f_1(x,y) & = 4x^2 + 4y^2 \\ f_2(x,y) & = (x-5)^2 + (y-5)^2 \end{cases}$$

$$\text{s.t. } \begin{cases} g_1(x,y) & = (x-5)^2 + y^2 \leq 25 \\ g_2(x,y) & = (x-8)^2 + (y+3)^2 \geq 7.7 \end{cases}$$

for $0 \leq x \leq 5$, $0 \leq y \leq 3$.

- **Chakong and Haimes function:** (Fig. **A-2**(b))

$$\text{Minimize: } \begin{cases} f_1(x,y) & = 2 + (x-2)^2 + (y-1)^2 \\ f_2(x,y) & = 9x + (y-1)^2 \end{cases}$$

$$\text{s.t. } \begin{cases} g_1(x,y) & = x^2 + y^2 \leq 225 \\ g_2(x,y) & = x - 3y + 10 \leq 0 \end{cases}$$

for $-20 \leq x, y \leq 20$.

- **Fonseca and Fleming function:** (Fig. **A-2**(c))

$$\text{Minimize: } \begin{cases} f_1(\boldsymbol{x}) & = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ f_2(\boldsymbol{x}) & = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \end{cases}$$

for $-4 \leq x_i \leq 4$, $1 \leq i \leq n$.

- **Test function 4 – [17]:** (Fig. **A-2**(d))

$$\text{Minimize: } \begin{cases} f_1(x,y) & = x^2 - y \\ f_2(x,y) & = -0.5x - y - 1 \end{cases}$$

$$\text{s.t.} \begin{cases} g_1(x,y) & = 6.5 - \frac{x}{6} - y \geq 0 \\ g_2(x,y) & = 7.5 - 0.5x - y \geq 0 \\ g_3(x,y) & = 30 - 5x - y \geq 0 \end{cases}$$

for $-7 \leq x, y \leq 4$.

- **Kursawe function:** (Fig. **A-2**(e))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = \sum_{i=1}^{2} \left[ -10 \exp\left( -0.2\sqrt{x_i^2 + x_{i+1}^2} \right) \right] \\ \\ f_2(\boldsymbol{x}) & = \sum_{i=1}^{3} \left[ |x_i|^{0.8} + 5\sin\left(x_i^3\right) \right] \end{cases}$$

for $-5 \leq x_i \leq 5$, $1 \leq i \leq 3$.

- **Schaffer function N. 1:** (Fig. **A-2**(f))

$$\text{Minimize:} \begin{cases} f_1(x) & = x^2 \\ f_2(x) & = (x-2)^2 \end{cases}$$

for $-A \leq x \leq A$. Values of $A$ from 10 to $10^5$ have been used successfully. Higher values of $A$ increase the difficulty of the problem.

- **Schaffer function N. 2:** (Fig. **A-2**(g))

$$\text{Minimize:} \begin{cases} f_1(x) & = \begin{cases} -x & \text{if } x \leq 1 \\ x-2 & \text{if } 1 < x \leq 3 \\ 4-x & \text{if } 3 < x \leq 4 \\ x-4 & \text{if } x > 4 \end{cases} \\ f_2(x) & = (x-5)^2 \end{cases}$$

for $-5 \leq x \leq 10$.

- **Poloni's two objective function:** (Fig. **A-2**(h))

$$\text{Minimize:} \begin{cases} f_1(x,y) & = \left[ 1 + (A_1 - B_1(x,y))^2 + (A_2 - B_2(x,y))^2 \right] \\ f_2(x,y) & = (x+3)^2 + (y+1)^2 \end{cases}$$

$$\text{where} \begin{cases} A_1 & = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ A_2 & = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ B_1(x,y) & = 0.5\sin(x) - 2\cos(x) + \sin(y) - 1.5\cos(y) \\ B_2(x,y) & = 1.5\sin(x) - \cos(x) + 2\sin(y) - 0.5\cos(y) \end{cases}$$

for $-\pi \leq x, y \leq \pi$.

- **Zitzler–Deb–Thiele's function N. 1:** (Fig. **A-2**(i))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = x_1 \\ f_2(\boldsymbol{x}) & = g(\boldsymbol{x})\, h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) \\ g(\boldsymbol{x}) & = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) & = 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}} \end{cases}$$

for $0 \leq x_i \leq 1$, $1 \leq i \leq 30$.

- **Zitzler–Deb–Thiele's function N. 2:** (Fig. **A-2**(j))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = x_1 \\ f_2(\boldsymbol{x}) & = g(\boldsymbol{x})\, h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) \\ g(\boldsymbol{x}) & = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) & = 1 - \left(\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}\right)^2 \end{cases}$$

for $0 \leq x_i \leq 1$, $1 \leq i \leq 30$.

- **Zitzler–Deb–Thiele's function N. 3:** (Fig. **A-2**(k))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = x_1 \\ f_2(\boldsymbol{x}) & = g(\boldsymbol{x})\, h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) \\ g(\boldsymbol{x}) & = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) & = 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}} - \left(\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}\right)\sin(10\pi f_1(\boldsymbol{x})) \end{cases}$$

for $0 \leq x_i \leq 1$, $1 \leq i \leq 30$.

- **Zitzler–Deb–Thiele's function N. 4:** (Fig. **A-2**(l))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = x_1 \\ f_2(\boldsymbol{x}) & = g(\boldsymbol{x})\, h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) \\ g(\boldsymbol{x}) & = 91 + \sum_{i=2}^{10} \left(x_i^2 - 10\cos(4\pi x_i)\right) \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) & = 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}} \end{cases}$$

for $0 \leq x_1 \leq 1$, $-5 \leq x_i \leq 5$, $2 \leq i \leq 10$.

- **Zitzler–Deb–Thiele's function N. 6:** (Fig. **A-2**(m))

$$\text{Minimize:} \begin{cases} f_1(\boldsymbol{x}) & = 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ f_2(\boldsymbol{x}) & = g(\boldsymbol{x})\, h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) \\ g(\boldsymbol{x}) & = 1 + 9\left[\frac{\sum_{i=2}^{10} x_i}{9}\right]^{0.25} \\ h(f_1(\boldsymbol{x}), g(\boldsymbol{x})) & = 1 - \left(\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}\right)^2 \end{cases}$$

for $0 \leq x_i \leq 1$, $1 \leq i \leq 10$.

- **Viennet function:** (Fig. **A-2**(o))

$$\text{Minimize:} \begin{cases} f_1(x,y) & = 0.5(x^2+y^2) + \sin(x^2+y^2) \\ f_2(x,y) & = \frac{(3x-2y+4)^2}{8} + \frac{(x-y+1)^2}{27} + 15 \\ f_3(x,y) & = \frac{1}{x^2+y^2+1} - 1.1\exp(-(x^2+y^2)) \end{cases}$$

for $-3 \leq x, y \leq 3$.

- **Osyczka and Kundu function:** (Fig. **A-2**(p))

$$\text{Min.:} \begin{cases} f_1(\boldsymbol{x}) & = -25(x_1-2)^2 - (x_2-2)^2 - (x_3-1)^2 - (x_4-4)^2 - (x_5-1)^2 \\ f_2(\boldsymbol{x}) & = \sum_{i=1}^{6} x_i^2 \end{cases}$$

$$\text{s.t.} \begin{cases} g_1(\boldsymbol{x}) & = x_1 + x_2 - 2 \geq 0 \\ g_2(\boldsymbol{x}) & = 6 - x_1 - x_2 \geq 0 \\ g_3(\boldsymbol{x}) & = 2 - x_2 + x_1 \geq 0 \\ g_4(\boldsymbol{x}) & = 2 - x_1 + 3x_2 \geq 0 \\ g_5(\boldsymbol{x}) & = 4 - (x_3-3)^2 - x_4 \geq 0 \\ g_6(\boldsymbol{x}) & = (x_5-3)^2 + x_6 - 4 \geq 0 \end{cases}$$

for $0 \leq x_1, x_2, x_6 \leq 10$, $1 \leq x_3, x_5 \leq 5$, $0 \leq x_4 \leq 6$.

- **CTP1 function (2 variables) – [33]:** (Fig. **A-2**(q))

$$\text{Minimize: } \begin{cases} f_1(x,y) & = x \\ f_2(x,y) & = (1+y)\exp\left(-\frac{x}{1+y}\right) \end{cases}$$

$$\text{s.t. } \begin{cases} g_1(x,y) & = \frac{f_2(x,y)}{0.858\exp(-0.541 f_1(x,y))} \geq 1 \\ g_1(x,y) & = \frac{f_2(x,y)}{0.728\exp(-0.295 f_1(x,y))} \geq 1 \end{cases}$$

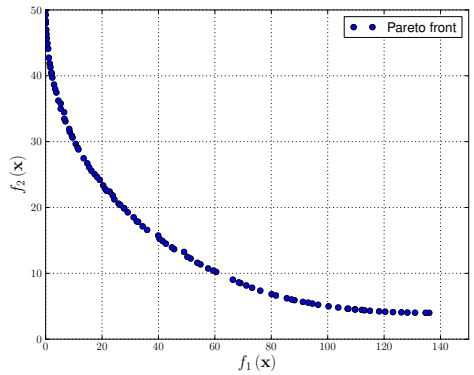
for $0 \leq x, y \leq 1$.

- **Constr–Ex problem – [33]:** (Fig. **A-2**(r))

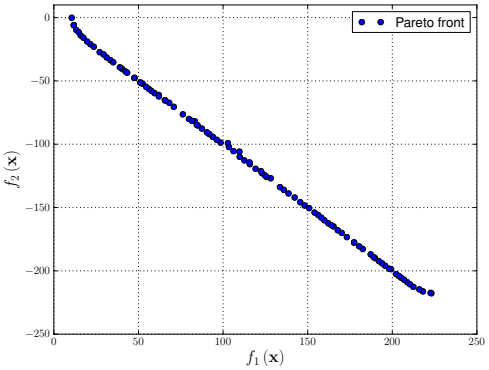
$$\text{Minimize: } \begin{cases} f_1(x,y) & = x \\ f_2(x,y) & = \frac{1+y}{x} \end{cases}$$

$$\text{s.t. } \begin{cases} g_1(x,y) & = y + 9x \geq 6 \\ g_1(x,y) & = -y + 9x \geq 1 \end{cases}$$

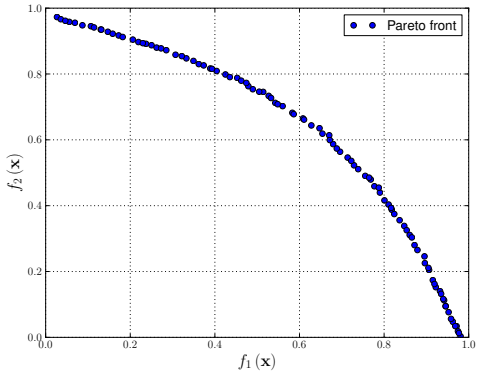
for $0.1 \leq x \leq 1$, $0 \leq y \leq 5$.



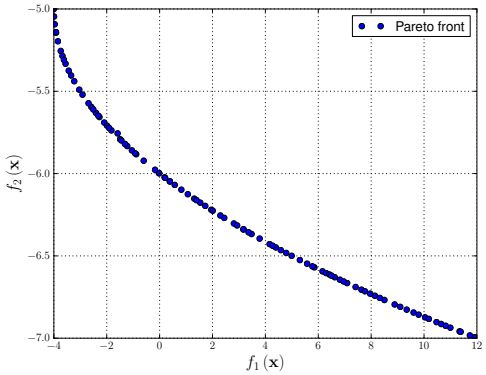

(a) Binh and Korn function



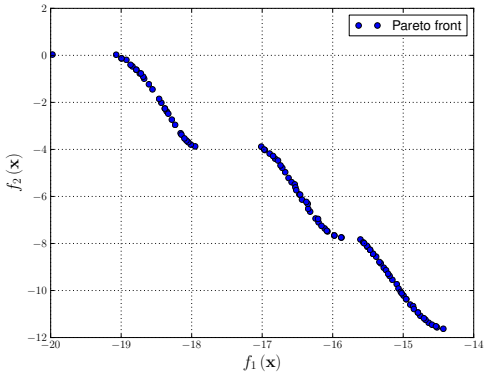

(b) Chakong and Haimes function



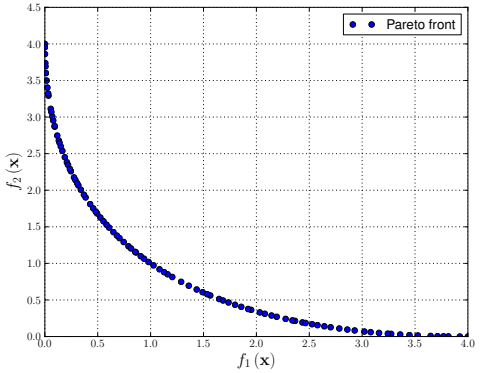

(c) Fonseca and Fleming function (3 variables)




(d) Test function 4 – [17]

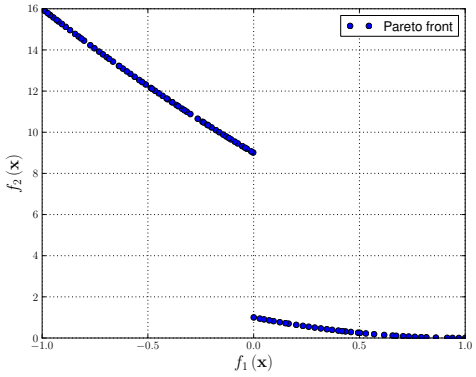



(e) Kursawe function

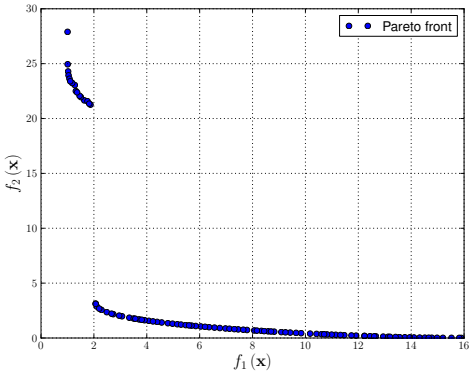



(f) Schaffer function N. 1

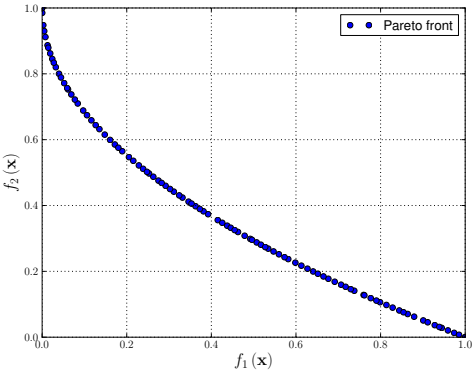
**Figure A-2**.: Test functions for multi–objective optimization



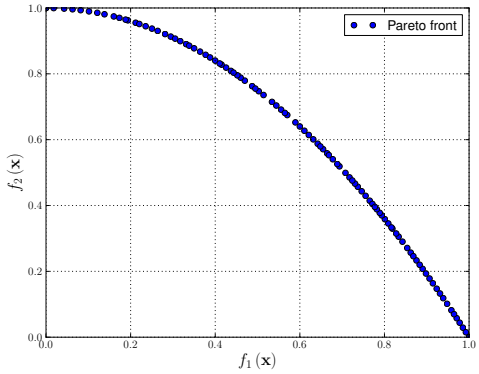

(g) Schaffer function N. 2



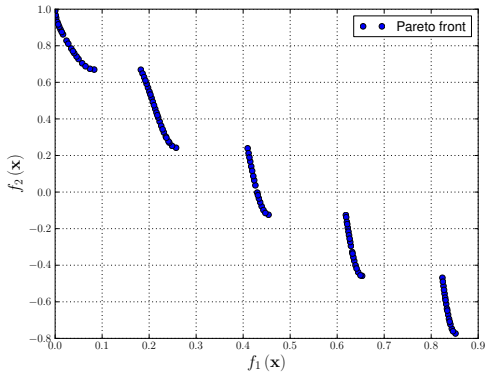

(h) Poloni's two objective function



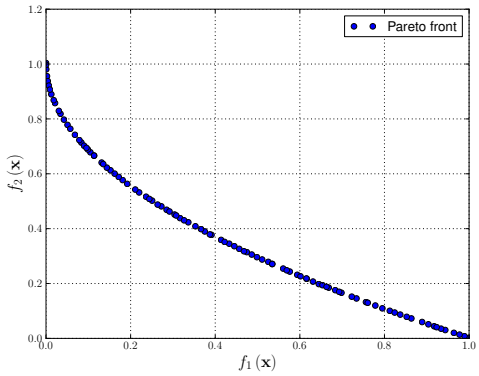

(i) Zitzler–Deb–Thiele's function N. 1




(j) Zitzler–Deb–Thiele's function N. 2

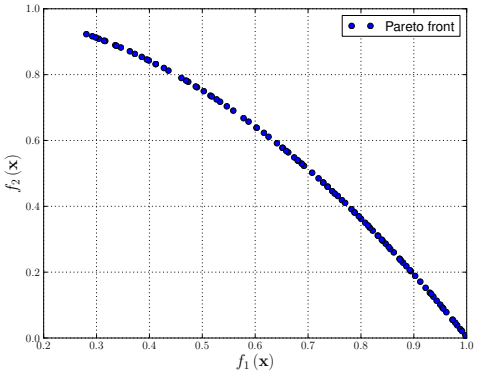



(k) Zitzler–Deb–Thiele's function N. 3

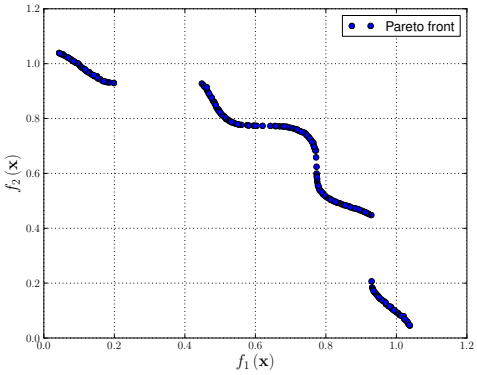



(l) Zitzler–Deb–Thiele's function N. 4

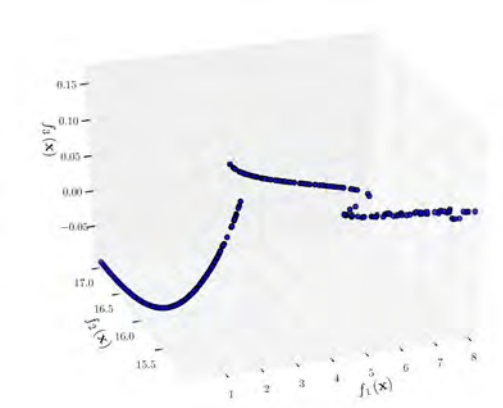
**Figure A-2**.: Test functions for multi–objective optimization



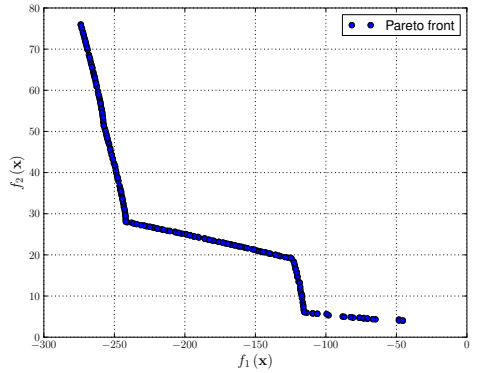

(m) Zitzler–Deb–Thiele's function N. 6



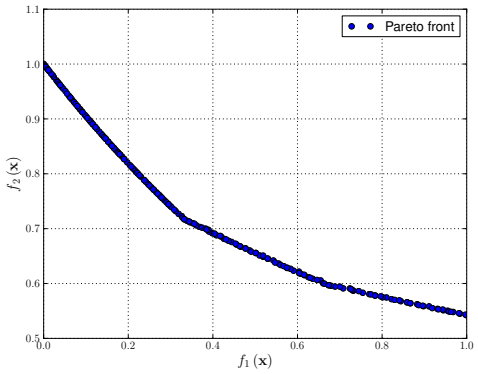

(n) Tanaka function



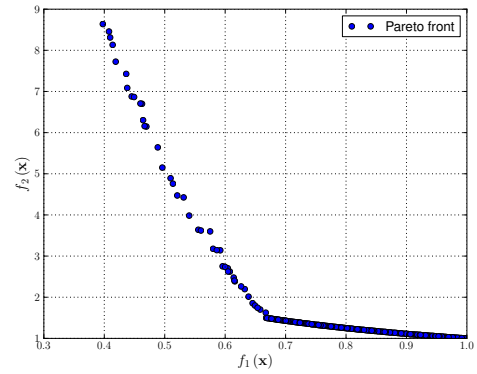

(o) Viennet function




(p) Osyczka and Kundu function




(q) CTP1 function (2 variables) – [33]




(r) Constr–Ex problem – [33]

**Figure A-2.**: Test functions for multi–objective optimization

# B. Research products

## B.1. Articles in high impact journals

- Ortiz, Gilberto A.; Alvarez, Diego A.; Bedoya-Ruíz, Daniel (2013). "Identification of Bouc-Wen type models using multi-objective optimization algorithms". In: *Computers & Structures*. Vol. 114-115. Pag. 121-132. January 2013. http://dx.doi.org/10.1016/j.compstruc.2012.10.016.

## B.2. Articles in conference proceedings

- Bedoya-Ruiz, Daniel; Alvarez, Diego A.; Ortiz, Gilberto A.; García Fergusson, Alberto (September 2012). "Cyclic behaviour of Precast Concrete Modules with Post-tensioned Unbounded Bars - Paper 3414". In: *15 WCEE - 15th World Conference on Earthquake Engineering*. Lisbon, Portugal. 24-28 September, 2012.

- Bedoya-Ruiz, Daniel; Bermúdez, Carlos A.; Alvarez, Diego A.; Ortiz, Gilberto A.; Escobar Sáenz, Juan Vicente (September 2012). "Cyclic behaviour of Prestressed precast concrete walls - Paper 3415". In: *15 WCEE - 15th World Conference on Earthquake Engineering*. Lisbon, Portugal. 24-28 September, 2012.

- Bedoya-Ruiz, Daniel; Alvarez Marín, Diego; Ortiz García, Gilberto (October 2012). "Comportamiento sísmico de muros de ferrocemento". pp. 529-543. In: *10th international symposium on ferrocement and thin reinforced cement composites*. Palacio de convenciones, la Habana, Cuba. 15-17 October, 2012. Edited by: Wainshtok Rivas, Hugo; Prada Seoane, Lázaro and Granda Castro, Iria. Editorial Obras. ISBN: 978-959-247-098-9.

- Bedoya-Ruiz, Daniel; Alvarez Marín, Diego; Hurtado Gómez, Jorge (October 2012). "Modelo dinámico no lineal para el comportamiento sísmico de viviendas de ferrocemento". pp. 545-554 (see acknowledgements). In: *10th international symposium on ferrocement and thin reinforced cement composites*. Palacio de convenciones, la Habana, Cuba. 15-17 October, 2012. Edited by: Wainshtok Rivas, Hugo; Prada Seoane, Lázaro and Granda Castro, Iria. Editorial Obras. ISBN: 978-959-247-098-9.

- Bedoya-Ruiz, Daniel; Hurtado Gómez, Jorge; Alvarez Marín, Diego (October 2012). "Nonlinear model and seismic vulnerability of ferrocement housing". pp. 555-562 (see acknowledgements). In: *10th international symposium on ferrocement and thin reinforced cement composites*. Palacio de convenciones, la Habana, Cuba. 15-17 October, 2012. Edited by: Wainshtok Rivas, Hugo; Prada Seoane, Lázaro and Granda Castro, Iria. Editorial Obras. ISBN: 978-959-247-098-9.

## B.3. Software

- For the identification process, the software developed by Deb [32] was employed, the source code can be found at Professor Deb website [32]. Using such source code as "main engine", the BW-type models of hysteresis and the four objective functions proposed above were implemented. The modified algorithm with the BW-type models of hysteresis can be found at `http://sourceforge.net/projects/boucwenbabernoo/`.

- There are programs developed in MATLAB® that were used to evaluate the performance of the algorithms used in this Thesis. The programs are posted online [75, 74].

# Bibliography


[1] Ajavakom, N. ; Ng, C.H. ; Ma, F.: Performance of nonlinear degrading structures: Identification, validation, and prediction. En: *Computers & Structures* 86 (2008), Nr. 7–8, p. 652–662. – ISSN 0045–7949

[2] Akazawa, T. ; Nakashima, M. ; Sakaguchi, O.: Simple model for simulating hysteretic behavior involving significant strain hardening - Paper 264. En: Sociedad Mexicana de Ingeniería (Ed.): *Proceedings of the 11th world conference on earthquake engineering* Vol. 1, 1996

[3] Arulampalam, M. S. ; Maskell, Simon ; Gordon, Neil: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. En: *IEEE Transactions on Signal Processing* 50 (2002), p. 174–188

[4] Atalay, M. B. ; Penzien, J.: The seismic behavior of critical regions of reinforced concrete components influenced by moment, shear and axial force / University of California, Berkeley. 1975 ( 75-19). – UCB/EERC technical report

[5] Baber, T. T. ; Noori, M. N.: Modeling general hysteresis behaviour and random vibration applications. En: *Journal of Vibration, Acoustics, Stress, and Reliability in Design* 108 (1986), Nr. 4, p. 411–420

[6] Baber, Thomas T. ; Noori, Mohammad N.: Random Vibration of Degrading, Pinching Systems. En: *Journal of Engineering Mechanics* 111 (1985), Nr. 8, p. 1010–1026

[7] Baber, Thomas T. ; Wen, Yi-Kwei: Random Vibration Hysteretic, Degrading Systems. En: *Journal of Engineering Mechanics* 107 (1981), Nr. 6, p. 1069–1087

[8] Bäck, Thomas: *Evolutionary Algorithms in Theory and Practice. Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Dortmund, Germany : Oxford University Press, 1996. – ISBN 0–19–509971–0

[9] Bahar, Arash ; Pozo, Francesc ; Acho, Leonardo ; Rodellar, José ; Barbat, Alex: Hierarchical semi-active control of base-isolated structures using a new inverse model of magnetorheological dampers. En: *Computers & Structures* 88 (2010), Nr. 7–8, p. 483–496

[10] Barbat, Álex H. ; Canet, J. M.: *Estructuras sometidas a acciones sísmicas: Cálculo por ordenador*. 2. Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE), 1994. – ISBN 8487–86710–3

[11] Barbato, G. ; Brondino, G. ; Galleto, M. ; Levi, R. *An automatic procedure for evaluation of mechanical parameters of metallic materials*

[12] Bedoya-Ruiz, Daniel ; Hurtado, Jorge E. ; Pujades, Lluis: Experimental and analytical research on seismic vulnerability of low-cost ferrocement dwelling houses. En: *Structure and Infrastructure Engineering* 6 (2010), Nr. 1-2, p. 55–62

[13] Bedoya-Ruiz, Daniel A.: *Estudio de resistencia y vulnerabilidad sísmicas de viviendas de bajo costo estructuradas con ferrocemento*, Universidad Politécnica de Catalunya, Ph.D. dissertation, 2005

[14] Bellman, R.E.: *Adaptive control processes: a guided tour*. Princeton, NJ : Princeton University Press, 1961 (Rand Corporation Research studies)

[15] Benson, H.: An algorithm for optimising over the weakly-efficient set. En: *European journal of operational research* 25 (1986), p. 192–199

[16] Van den Bergh, Frans: *An Analysis of Particle Swarm Optimizers*, University of Pretoria, Pretoria, Ph.D. dissertation, 2001

[17] Binh, Thanh: A multiobjective evolutionary algorithm. The study cases / Institute for Automation and Communication. Barleben, Germany, 1999. – Technical report

[18] Binh, Thanh ; Korn, Ulrich: MOBES: A Multiobjective Evolution Strategy for Constrained Optimization Problems. En: Mendel 97 (Ed.): *Proceedings of the Third International Conference on Genetic Algorithms*. Czech Republic, 1997, p. 176–182

[19] Bouc, R.: Forced vibration of mechanical systems with hysteresis. En: Djadkov, Sergej (Ed.): *Proceedings of the Fourth International Conference on Nonlinear Oscillation*. Prague, Czechoslovakia, 1967, p. p. 315

[20] Bouc, R.: Modèle mathématique d'hystérésis. En: *Acustica* 24 (1971), p. 16–25

[21] Brokate, M. ; Sprekels, J: Existence and optimal control of mechanical processess with hysteresis in viscous solids. En: *Journal of applied mathematics* 43 (1989), p. 219–229

[22] CESMD: *Center for Engineering Strong Motion Data*. Available at http://www.strongmotioncenter.org/cgi-bin/CESMD/search1.pl. 4 2012. – Northridge earthquake (1994), Santa Monica. Station 24538

[23] Charalampakis, A.E. ; Dimou, C.K.: Identification of Bouc-Wen hysteretic systems using particle swarm optimization. En: *Computers & Structures* 88 (2010), Nr. 21-22, p. 1197–1205. – ISSN 0045–7949

[24] Charalampakis, A.E. ; Koumousis, V.K.: Identification of Bouc-Wen hysteretic systems by a hybrid evolutionary algorithm. En: *Journal of Sound and Vibration* 314 (2008), Nr. 3-5, p. 571–585. – ISSN 0022–460X

[25] Chatzi, Eleni N. ; Smyth, Andrew W.: The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. En: *Structural Control and Health Monitoring* 16 (2009), Nr. 1, p. 99–123. – ISSN 1545–2263

[26] Chatzi, Eleni N. ; Smyth, Andrew W. ; Masri, Sami F.: Experimental application of on–line parametric identification for nonlinear hysteretic systems with model uncertainty. En: *Structural Safety* 32 (2010), Nr. 5, p. 326–337. – ISSN 0167–4730

[27] Chiroiu, V. ; Munteanu, L.: Identification of hysteretic behaviour of materials by using genetic algorithms. En: Mastorakis, Nikos E. (Ed.) ; Croitoru, Anca (Ed.) ; Balas, Valentina E. (Ed.) ; Son, Eduard (Ed.) ; Mladenov, Valeri (Ed.): *Proceedings of the 10th WSEAS international conference on automation & information* Vol. 1, 2009. – ISSN 1790–5117, p. 46–51

[28] Chopra, Anil K.: *Dynamics of Structures. Theory and Applications to Earthquake Engineering*. 3. Berkeley, California, USA : Prentice-Hall Inc, 2006. – ISBN 0–13–156174–X

[29] Clough, R. W.: Effect of stiffness degradation on earthquake ductility requirement / University of California, Berkeley. 1966 ( 6614). – Technical report

[30] Clough, Ray W. ; Penzien, Joseph: *Dynamics of Structures*. McGraw–Hill Companies, 1975. – ISBN 0–07–011392–0

[31] Coello Coello, Carlos A. ; Van Veldhuizen, David A. ; Lamont, Gary B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, USA : Kluwer Academic/Plenum Publishers, 2002. – ISBN 0–306–46762–3

[32] Deb, K.: *Software for multi-objective NSGA-II code in C*. Available at http://www.iitk.ac.in/kangal/codes.shtml. 7 2011. – Revision 1.1.6

[33] Deb, Kalyanmoy: *Multi-Objective Optimization using Evolutionary Algorithms*. Kanpur, India : John Wiley & Sons Inc., 2001. – ISBN 0–471–87339–X

[34] Deb, Kalyanmoy ; Pratap, Amrit ; Agarwal, Sameer ; Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. En: *IEEE transactions on Evolutionary Computation* 6 (2002), Nr. 2, p. 182–197. – ISSN 1089–778X

[35] Doucet, A. ; Johansen, A. M.: A Tutorial on Particle filtering and smoothing: Fifteen years later. En: Crisan, D. (Ed.) ; Rozovsky, B. (Ed.): *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011

[36] Eiben, A. E. ; Smith, J. E.: *Introduction to evolutionary computing*. Berlin : Springer, 2003

[37] Foliente, G. C. ; Singh, M. P. ; Noori, M. N.: Equivalent linearization of generally pinching hysteretic, degrading systems. En: *Earthquake Engineering and Structural Dynamics* 25 (1996), Nr. 6, p. 611–629

[38] Foliente, Greg C.: Hysteresis Modeling of Wood Joints and Structural Systems. En: *Journal of Structural Engineering* 121 (1995), Nr. 6, p. 1013–1022

[39] Fonseca, C.M. ; Fleming, P.J.: Multiobjective genetic algorithms. En: *IEEE Colloquium on Genetic Algorithms for Control Systems Engineering*. London, UK, 5 1993, p. 6/1 –6/5

[40] Fukada, Y.: A study on the restoring force characteristics of reinforced concrete buildings. En: *Proceedings of the Kanto district symposium of AIJ, Tokyo, Japan* Vol. 40, 1969, p. 121–124

[41] Garrido, Ruben ; Rivero-Angeles, Francisco J.: Hysteresis and parameter estimation of MDOF systems by a continuous-time least squares method. En: *Journal of Earthquake Engineering* 10 (2006), Nr. 2, p. 237–264

[42] Gerolymos, Nikos ; Gazetas, George: Development of Winkler model for static and dynamic response of caisson foundations with soil and interface nonlinearities. En: *Soil Dynamics and Earthquake Engineering* 26 (2006), Nr. 5, p. 363–376

[43] Giuclea, M. ; Sireteanu, T. ; Mitu, A. M.: Use of genetic algorithms for fitting the Bouc-Wen model to experimental hysteretic curves. En: *Revue Roumaine des Sciences Techniques - Serie De Mecanique Appliquee* 54 (2009), Nr. 1, p. 3–10

[44] Gomis-Bellmunt, Oriol ; Ikhouane, Fayçal ; Montesinos-Miracle, Daniel: Control of a piezoelectric actuator considering hysteresis. En: *Journal of Sound and Vibration* 326 (2009), Nr. 3–5, p. 383–399

[45] Grewal, Mohinder S. ; Andrews, Angus P.: *Kalman filtering: Theory and practice using Matlab*. 3. John Wiley & Sons Inc., 2008. – ISBN 0–470–17366–1

[46] Guglielmino, Emanuele ; Sireteanu, Tudor ; Stammers, Charles W. ; Gheorghe, Ghita ; Giuclea, Marius: *Semi-active Suspension Control. Improved Vehicle Ride and Road Friendliness*. London, UK : Springer, 2008. – ISBN 1–848–00230–0

[47] Haupt, Randy L. ; Haupt, Sue E.: *Practical Genetic Algorithms*. 2. New Jersey, USA : John Wiley & Sons Inc, 2004. – ISBN 0–471–45565–2

[48] Haykin, S.: *Kalman filtering and neural networks*. 1. John Wiley & Sons Inc., 2001. – ISBN 0–471–36998–5

[49] Holland, John H.: Outline for a Logical Theory of Adaptive Systems. En: *J. ACM* 9 (1962), p. 297–314. – ISSN 0004–5411

[50] Holland, John H.: *Adaptation in natural and artificial systems*. Ann Arbor, MI, USA : Michigan University Press, 1975

[51] Hurtado, Jorge E.: *Introducción a la dinámica de estructuras*. Universidad Nacional de Colombia, 8 2000. – ISBN 958–9322581

[52] Ibarra, L. F. ; Medina, R. A. ; Krawinkler, H.: Hysteretic models that incorporate strength and stiffness deterioration. En: *Earthquake engineering and structural dynamics* 34 (2005), Nr. 1, p. 1489–1511

[53] Ikhouane, F. ; Rodellar, J.: *Systems with hysteresis: Analysis, identification and control using the Bouc-Wen model*. Barcelona, Spain : John Wiley & Sons Ltd., 2007. – ISBN 0–470032367

[54] Ismail, M. ; Rodellar, J. ; Ikhouane, F.: An innovative isolation device for aseismic design. En: *Engineering Structures* 32 (2010), Nr. 4, p. 1168 – 1183

[55] Ismail, Mohammed ; Ikhouane, Fayçal ; Rodellar, José: The Hysteresis Bouc-Wen Model, a Survey. En: *Archives of Computational Methods in Engineering* 16 (2009), Nr. 2, p. 161–188

[56] Jang, Jyh-Shing R. ; Sun, Chuen-Tsai ; Mizutani, Eiji: *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. New Jersey, USA : Prentice Hall Inc., 1997. – ISBN 0–132–61066–3

[57] Julier, Simon J. ; Uhlmann, Jeffrey K.: A New Extension of the Kalman Filter to Nonlinear Systems. En: SPIE (Ed.): *Proceedings of AeroSense: the 11th international symposium on aerospace/defense sensing, simulation and control*. Orlando, 1997, p. 182–193

[58] Kandepu, Rambabu ; Foss, Bjarne ; Imsland, Lars: Applying the unscented Kalman filter for nonlinear state estimation. En: *Journal of Process Control* 18 (2008), Nr. 7-8, p. 753–768

[59] Kennedy, J. ; Eberhart, R. C.: Particle swarm optimization. En: Neural Networks council, IEEE (Ed.): *Proceedings of IEEE international conference on neural networks IV*. Perth, Australia, 1995, p. 1942–1948

[60] Koumousis, V. K. ; Katsaras, T. P.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. En: *IEEE transactions on Evolutionary Computation* 10 (2006), Nr. 1, p. 19–28

[61] Kuczmann, M.: Dynamic Preisach hysteresis model. En: *Journal of advanced research in physics* 1 (2010), Nr. 1, p. 1–5

[62] Kunnath, Sashi K. ; Mander, John B. ; Fang, Lee: Parameter identification for degrading and pinched hysteretic structural concrete systems. En: *Engineering Structures* 19 (1997), Nr. 3, p. 224–232. – ISSN 0141–0296

[63] Kwok, N.M. ; Ha, Q.P. ; Nguyen, M.T. ; Li, J. ; Samali, B.: Bouc-Wen model parameter identification for a MR fluid damper using computationally efficient GA. En: *ISA Transactions* 46 (2007), Nr. 2, p. 167–179

[64] Kyprianou, A. ; Worden, K. ; Panet, M.: Identification of hysteretic systems using the differential evolution algorithm. En: *Journal of Sound and Vibration* 248 (2001), Nr. 2, p. 289–314. – ISSN 0022–460X

[65] Li, S. J. ; Suzuki, Y. ; Noori, M.: Identification of hysteretic systems with slip using bootstrap filter. En: *Mechanical systems and signal processing* 18 (2004), Nr. 4, p. 781–795

[66] Li, S. J. ; Suzuki, Y. ; Noori, M.: Improvement of parameter estimation for nonlinear hysteretic systems with slip by a fast Bayesian bootstrap filter. En: *International journal of non-linear mechanics* 39 (2004), Nr. 9, p. 1435–1445

[67] Li, Zhengying ; Albermani, Faris ; Chan, Ricky W. ; Kitipornchai, S.: Pinching hysteretic response of yielding shear panel device. En: *Journal of Engineering Structures* 33 (2011), Nr. 3, p. 993–1000. – ISSN 0141–0296

[68] Lubliner, J.: *Plasticity theory*. New York : Dover Publications, 2008. – ISBN 978–0486462905

[69] Luenberger, David G.: *Optimization by Vector Space Methods*. Stanford, California, USA : John Wiley & Sons, Inc., 1969. – ISBN 0471–55359–X

[70] In: Messac, Achille ; Mullur, Anoop A.: *Multiobjective Optimization: Concepts and Methods*. Toh Tuck Link, Singapore : World Scientific Publishing Co., 2007, p. 121–147. – ISBN 978–981–256–962–2

[71] Miettinen, Kaisa: *Nonlinear multi-objective optimization*. Massachusetts, USA : Kluwer Academic Publishers, 1999. – ISBN 0–7923–8278–1

[72] Oldenhuis, R. *Many test functions for global optimizers*. Available at http://www.mathworks.com/matlabcentral/fileexchange. 7 2011

[73] Oldfield, Matthew ; Ouyang, Huajiang ; Mottershead, John E.: Simplified models of bolted joints under harmonic loading. En: *Computers & Structures* 84 (2005), Nr. 1–2, p. 25–33

[74] Ortiz, Gilberto A. *Evolution Strategies (ES)*. Available at http://www.mathworks.com/matlabcentral/fileexchange. 7 2012

[75] Ortiz, Gilberto A. *Multi-objective optimization using ES as Evolutionary Algorithm*. Available at http://www.mathworks.com/matlabcentral/fileexchange. 7 2012

[76] Park, Y. J. ; Reinhorn, A. M. ; Kunnath, S. K.: IDARC: Inelastic Damage Analysis of Reinforced Concrete frame-shear-wall structures / State University of New York at Buffalo. 1987 ( 87-0008). – Technical report NCEER

[77] Park, Y. J. ; Wen, Y. K. ; Ang, A. H-S.: Random vibration of hysteretic systems under bi-directional ground motions. En: *Earthquake Engineering and Structural Dynamics* 14 (1986), Nr. 4, p. 543–557

[78] Pires, J.A.: Stochastic seismic response analysis of soft soil sites. En: *Nuclear Engineering and Design* 160 (1996), Nr. 3, p. 363–377. – ISSN 0029–5493

[79] Preisach, F.: Über die magnetische Nachwirkung. En: *Zeitschrift für Physik A Hadrons and Nuclei* 94 (1935), p. 277–302. – ISSN 0939–7922

[80] Ramberg, W. ; Osgood, W. R.: Description of stress–strain curves by three parameters / National Advisory Committee for Aeronautics. 1943 ( 902). – Technical note

[81] Rechenberg, I.: Cybernetic solution path of an experimental problem / Royal Air Force Establishment. Farnborough, Hants., UK, 8 1965. – Library transaltion No. 1122

[82] Särkkä, Simo: *Recursive Bayesian inference on stochastic differential equations*, Helsinki University of Technology, Ph.D. dissertation, 2006

[83] Schaffer, J. D.: *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, Vanderbilt University, Ph.D. dissertation - Unpublished, 1984

[84] Schwefel, H. P.: Projekt MHD-Staustrahlrohr: Experimentelle Optimierung einer Zweiphasendüse, Teil I / AEG Forschungsinstitut. Berlin, Germany, 10 1968. – Technischer Bericht 11.034/68

[85] Schwefel, H. P.: Collective phenomena in evolutionary systems. En: Checkland, Peter (Ed.): *Preprints of the 31st Annual Meeting of the International Society for General System Research* Vol. 2. Budapest, 6 1987, p. 1025–1033

[86] Sivaselvan, M. V. ; Reinhorn, A. M.: Hysteretic Models for Cyclic Behavior of Deteriorating Inelastic Structures / Multidisciplinary Center for Earthquake Engineering Research. 1999 ( 99–1502). – Technical Report MCEER–99–0018

[87] Song, Junho ; Der Kiureghian, Armen: Generalized Bouc-Wen Model for Highly Asymmetric Hysteresis. En: *Journal of Engineering Mechanics* 132 (2006), Nr. 6, p. 610–618. – (See also discussion by F. Ihkouane and F. Pozo and L. Acho. Journal of Engineering Mechanics. ASCE. Volume 134, Issue 5, pp. 438–439, May 2008)

[88] Srinivas, N. ; Deb, Kalyanmoy: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. En: *Evolutionary Computation* 2 (1994), p. 221–248

[89] Wan, E.A. ; Van Der Merwe, R.: The unscented Kalman filter for nonlinear estimation. En: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS–SPCC. The IEEE 2000*, 2000, p. 153–158

[90] Wan, E.A. ; Van Der Merwe, R.: The Unscented Kalman Filter. En: Haykin, S. (Ed.): *Kalman filtering and neural networks*. John Wiley and Sons, 2001

[91] Wang, C. H. ; Wen, Y. K.: Reliability and redundancy of pre-Northridge low-rise steel building under seismic action / Univ. Illinous at Urbana-Champaign. Champaign, Ill., 1998 ( Rep No. UILU-ENG-99-2002). – Informe de Investigación

[92] Wang, Chi-Hsiang ; Chang, Shuenn-Yih: Development and Validation of a Generalized Biaxial Hysteresis Model. En: *Journal of Engineering Mechanics* 133 (2007), Nr. 2, p. 141–152

[93] Wen, Yi-Kwei: Method for random vibration of hysteretic systems. 102 (1976), Nr. 2, p. 249–263

[94] Worden, K. ; Hensman, J.J.: Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference. En: *Mechanical Systems and Signal Processing* 32 (2012), Nr. 0, p. 153 – 169

[95] Worden, K. ; Manson, G.: On the identification of hysteretic systems. Part I: Fitness landscapes and evolutionary identification. En: *Mechanical Systems and Signal Processing* 29 (2012), Nr. 0, p. 201 – 212

[96] Wu, Meiliang ; Smyth, Andrew S.: Real-time parameter estimation for degrading and pinching hysteretic models. En: *International Journal of Non-Linear Mechanics* 43 (2008), Nr. 9, p. 822–833. – ISSN 0020–7462

[97] Xie, S. L. ; Zhang, Y. H. ; Chen, C. H. ; Zhang, X. N.: Identification of nonlinear hysteretic systems by artificial neural networks (in press). En: *Mechanical Systems and Signal Processing* (2012)

[98] Xie, Zongbo ; Feng, Jiuchao: Real-time nonlinear structural system identification via iterated unscented Kalman filter. En: *Mechanical Systems and Signal Processing* 28 (2012), Nr. 0, p. 309–322. – ISSN 0888–3270

[99] Yang, Y. ; Ma, F.: Constrained Kalman Filter for Nonlinear Structural Identification. En: *Journal of Vibration and Control* 9 (2003), p. 1343–1357

[100] Ye, Meiying ; Wang, Xiaodong: Parameter estimation of the Bouc-Wen hysteresis model using particle swarm optimization. En: *Smart Materials and Structures* 16 (2007), Nr. 6, p. 2341–2349

[101] Zhang, H. ; Foliente, G. C. ; Yang, Y. ; Ma, F.: Parameter identification of inelastic structures under dynamic loads. En: *Earthquake Engineering and Structural Dynamics* 31 (2002), p. 1113–1130

[102] Zitzler, E. ; Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. En: *Evolutionary Computation, IEEE Transactions on* 3 (1999), 11, Nr. 4, p. 257 –271

[103] Zitzler, Eckart ; Laumanns, Marco ; Thiele, Lothar: SPEA2: Improving the Strength Pareto Evolutionary Algorithm / Swiss Federal Institute of Technology (ETH). Zurich, Switzerland, 2001. – Technical report 103

# Declaración

Me permito afirmar que he realizado la presente tesis de manera autónoma y con la única ayuda de los medios permitidos y no diferentes a los mencionados en la propia tesis. Todos los pasajes que se han tomado de manera textual o figurativa de textos publicados y no publicados, los he reconocido en el presente trabajo. Ninguna parte del presente trabajo se ha empleado en ningún otro tipo de tesis.

Manizales, Colombia, 15.04.2013

Gilberto Alejandro Ortiz García